



**Micael Alexandre
Saimeiro Pêgo**

**Desenvolvimento de plataforma para teste de
estratégias de controlo de equipamento para
aquecimento de águas domésticas**



**Micael Alexandre
Saimeiro Pêgo**

**Desenvolvimento de plataforma para teste de
estratégias de controlo de equipamento para
aquecimento de águas domésticas**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Mecânica, realizada sob orientação científica de Jorge Augusto Fernandes Ferreira, Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro, e de Vítor António Ferreira da Costa, Professor Associado com Agregação do Departamento de Engenharia Mecânica da Universidade de Aveiro.

O júri / The jury

Presidente / President

Doutor Nelson Amadeu Dias Martins

Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro

Vogais / Committee

Mestre Ricardo Miguel Santos Heleno

Engenheiro da Bosch Termotecnologia, Sa (Arguente)

Doutor Jorge Augusto Fernandes Ferreira

Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro (Orientador)

Agradecimentos / Acknowledgements

Aos meus pais, irmã, e a todos os restantes familiares que sempre depositaram apoio e confiança em mim, e me disponibilizaram todos os meios possíveis para alcançar os meus objetivos.

Aos amigos que estiveram do meu lado desde o primeiro dia.

Ao professor Jorge Ferreira pelo apoio incondicional ao longo do trabalho, mostrando-se sempre disponível.

Ao professor Vítor Costa que sempre mostrou preocupação e disponibilidade no decorrer do trabalho.

À Bosch Termotecnologia, nomeadamente ao Engenheiro David Guilherme, pelo interesse no tema e apoio para que este trabalho pudesse chegar a bom porto.

Palavras-chave

Hardware-in-the-loop-simulation; bancada de ensaios; controlo de sistemas térmicos; simulação em tempo real; desenvolvimento de sistemas.

Resumo

As estratégias de controlo para equipamentos de águas domésticas têm vindo a tornar-se cada vez mais complexas, não só pela necessidade de modernização de equipamentos, mas também devido aos constrangimentos do "*time-to-market*" a que a indústria produtiva está sujeita. Técnicas como o hardware-in-the-loop têm um papel fundamental para o teste de controladores, de forma a identificar problemas numa etapa inicial do projeto. Em parceria com a Bosch Termotecnologia, propôs-se um layout de bancada de ensaios que permita servir de apoio nas várias fases de projeto da empresa no teste de controladores, não só ao nível de hardware real como também simulado. Com o layout aprovado, efetuou-se o dimensionamento de todos os componentes de forma a permitir interação entre os mesmos para controlo, aquisição de dados e simulação. Por fim, foi construída uma pequena bancada de ensaios equipada com o hardware e software apropriado para os testes e foi realizada a respetiva validação dos resultados através de um caso de estudo. Iniciou-se também o projeto com os requisitos necessários para implementação do conceito num esquentador da Bosch Termotecnologia.

Keywords

Hardware-in-the-loop-simulation; bench tests; heater system's control; real time simulation; systems development

Abstract

The controll strategies for domestic water heater equipments have become increasingly complex, not only for the modernization of equipment, but also due to the constraints of time-to-market of the productive industry suffer. Technics like hardware-in-the-loop-simulation have been used for controller tests in order to identify problems in a initial phase of project. With collaboration of Bosch Termotechnology, it was proposed a bench tests layout that gives support in the multiple phases of controllers test, not only in real hardware as simulated too. With the layout validated, all components were designed in such a way that was possible the interaction between them to control, to data acquisition and simulation. Finally, a small test bench was built with the hardware and software for the tests and the results were validated through a case study. The project was initiated the project with the necessary requirements for the implementation of the bench test concept in a Bosch Termotechnology water heater.

Conteúdo

Lista de Tabelas	v
Lista de Figuras	vii
Lista de Acrónimos	ix
1 Introdução	1
1.1 Enquadramento	1
1.2 Apresentação da problemática	2
1.3 Objetivos	2
1.4 Guia de leitura	3
2 Revisão do Estado da Arte	5
2.1 Hardware in the loop simulation	5
2.2 Revisão tecnológica para HILS	6
2.2.1 <i>MATLAB/Simulink</i>	7
2.2.2 Real-Time Workshop	7
2.2.3 xPC Target	7
2.2.4 VxWorks	8
2.2.5 Real-Time Windows Target	8
2.2.6 RT-LAB	9
2.2.7 <i>software</i> dSPACE	9
2.2.8 National Instruments - PCI	10
2.2.9 Análise comparativa entre <i>software</i> e <i>hardware</i>	10
2.3 Bancadas de ensaio	12
2.3.1 Bancada para estudo de comportamento de esquentador	12
2.3.2 Projeto de bancada para a Vulcano	12
2.3.3 Simulação de um sistema de arrefecimento de um motor	13
2.3.4 Desenvolvimento de bancada para sistema de refrigeração	13
2.3.5 Controlador de velocidade de motores recorrendo ao <i>MATLAB/Simulink</i> e dSPACE	14
2.3.6 Teste a uma bomba de água de um sistema térmico	14
2.3.7 Balanço acerca das bancadas de ensaios	15
3 Conceito de bancada de ensaios	17
3.1 Proposta de conceito de <i>layout</i>	17
3.2 Funcionalidades do conceito	18

3.3	Características técnicas	21
3.3.1	Escolha de <i>hardware</i> e <i>software</i> para simulação e monitorização . .	21
3.3.2	Condicionamento de sinal	22
3.3.3	Comutação entre modos de operação	22
4	Caso de estudo 1	25
4.1	Definição do caso de estudo 1	25
4.1.1	Descrição da bancada	26
4.2	Modelação do Caso de Estudo 1	27
4.3	Controlador virtual	29
4.4	Comutação entre modos de utilização da bancada	31
4.5	Condicionamento do sinal	33
4.5.1	Drive de potência	34
4.5.2	Condicionamento do sinal para o sensor de temperatura	35
4.5.3	Condicionamento do sinal para o sensor de nível	36
4.6	Adaptações no Simulink	37
4.6.1	Adaptação dos sinais analógicos	37
4.6.2	Adaptação dos sinais digitais	38
4.6.3	Comutação através do modelo virtual	39
4.7	Interface e monitorização no ControlDesk	40
4.8	Programação do arduino	40
4.9	Montagem experimental	42
4.10	Procedimento experimental geral	45
4.11	Parametrização do modelo virtual	46
4.11.1	Testes de comportamento	47
4.12	Projeto do controlador	50
4.13	Testes de controlo	51
4.13.1	Teste do controlador virtual no modelo virtual	52
4.13.2	Teste do controlador virtual no modelo real	54
4.13.3	Teste do controlador real no modelo virtual	55
4.13.4	Teste do controlador real no modelo real	57
4.13.5	Segundo teste dos controladores no modelo real	57
4.14	Análise dos dados do caso de estudo 1	58
5	Caso de estudo 2	61
5.1	Definição do caso de estudo 2	61
5.2	Condicionamento de sinal	63
5.2.1	Válvula bypass	64
5.2.2	Válvula de água	65
5.2.3	Sensor de Temperatura NTC	67
5.2.4	Válvula de gás	69
5.2.5	Ventilador	69
5.3	Comutação entre simulações	71
5.4	Adaptações no <i>Simulink</i>	72
5.4.1	Exemplo de adaptação	72
5.5	Procedimento experimental	73
5.5.1	Modelo virtual	74

5.5.2	Interação entre modelo real e virtual	74
5.5.3	Interação entre a placa de controlo e o modelo virtual	74
5.5.4	Interação entre placa de controlo e protótipo	75
5.6	Considerações sobre o caso de estudo 2	75
6	Conclusões	77
6.1	Sugestões de Trabalhos futuros	78
Anexos		83
A	Projeto elétrico do caso de estudo 1	85
B	Código implementado no arduino	97
C	Interface gráfica do ControlDesk	105
D	Projeto elétrico do caso de estudo 2	109

Lista de Tabelas

2.1	Tabela resumo de <i>hardware</i> e <i>software</i> para simulação em tempo real . . .	12
4.1	Tabela de verdade para os contactos de comutação	32
4.2	Mapa de Karnaugh da saída 1	32
4.3	Mapa de Karnaugh da saída 2	32
4.4	Mapa de Karnaugh da saída 3	33
4.5	Gama de sinais de alimentação de saída/entrada	34
4.6	Lista de material associado aos componentes elétricos, de atuação e sensorização	43
4.7	Resumo das propriedades constantes usadas no sistema virtual e controlador	47
4.8	Valores de PID determinados por tentativa-erro.	50
4.9	Parâmetros alterados no modelo virtual, com base nas medições experimentais, para a simulação	52
5.1	Resumo do levantamento dos requisitos elétricos de cada componente . . .	64

Lista de Figuras

2.1	Possibilidades de interação numa experiência HILS em sistemas de controlo (Adaptado de [5])	6
2.2	Possibilidades de interação entre <i>software</i> e <i>hardware</i>	11
3.1	Esquema da proposta de layout esquemática da bancada de ensaios	18
3.2	Modo 1 de funcionamento da bancada de ensaios	19
3.3	Modo 2 de funcionamento da bancada de ensaios	19
3.4	Modo 3 de funcionamento da bancada de ensaios	20
3.5	Modo 4 de funcionamento da bancada de ensaios	20
3.6	Proposta de layout em 3D da bancada de ensaios	21
3.7	Esquema representativo com módulos de condicionamento e contactos	23
4.1	Representação de um controlador com <i>feedback</i> e <i>feedforward</i>	26
4.2	Esquema com o <i>layout</i> representativo do caso de estudo 1	27
4.3	Representação das variáveis envolvente nas perdas por condução e convecção	29
4.4	Representação de um controlador com <i>feedback</i> e <i>feedforward</i> (Adaptado de [26])	30
4.5	Esquema de comutação entre simulações	31
4.6	Circuito de acionamento dos atuadores de potência para a resistência e para a bomba de água	34
4.7	Esquema de amplificador diferencial do sinal de temperatura do LM35	35
4.8	Esquema do divisor da tensão aplicada à saída do sensor de nível	36
4.9	Esquema elétrico geral de funcionamento do caso de estudo	37
4.10	Conjunto de blocos para adaptação e envio do sinal de temperatura do reservatório pela saída analógica 2.	38
4.11	Conjunto de blocos para receção e adaptação do sinal de temperatura do reservatório pela entrada analógica 6.	38
4.12	Conjunto de blocos para receção e envio do sinal PWM de potência da resistência	39
4.13	Esquema de comutação entre simulações no Simulink	40
4.14	Representação do <i>arduino uno</i> com as ligações utilizadas no caso de estudo	41
4.15	Montagem de todo o sistema real, virtual e controlador real para os testes	44
4.16	Montagem do sistema real correspondente ao reservatório e respetivos atuadores e sensores	44
4.17	Montagem com os circuitos de condicionamento, <i>arduino</i> e <i>hardware</i> do sistema virtual	45

4.18	Evolução da temperatura com 50 % da potência máxima para o sistema real e virtual	48
4.19	Evolução da temperatura com 75 % da potência máxima para o sistema real e virtual	48
4.20	Evolução da temperatura com 100% da potência máxima para o sistema real e virtual	49
4.21	Teste a cada um dos controladores calculados P, PI e PID	51
4.22	Teste do controlador virtual no sistema virtual com Tsetup de 37°C.	53
4.23	Teste do controlador virtual no sistema virtual com Tsetup de 37°C, com maior foco na zona de estabilização.	53
4.24	Teste do controlador virtual no sistema real com Tsetup de 37°C.	54
4.25	Teste do controlador virtual no sistema real com Tsetup de 37°C, com maior foco na zona de estabilização.	55
4.26	Teste do controlador virtual e do <i>arduino</i> no sistema virtual com Tsetup de 37°C.	56
4.27	Teste do controlador virtual e do <i>arduino</i> no sistema virtual com Tsetup de 37°C, com maior foco na zona de estabilização.	56
4.28	Teste do <i>arduino</i> a controlar o sistema real com Tsetup de 37°C.	57
4.29	Teste do <i>arduino</i> e do controlador virtual no sistema real com Tsetup de 37°C.	58
5.1	Esquema exemplificativo dos componentes distribuídos pelo esquentador	62
5.2	Esquema demonstrativo das ligações pretendidas para o caso de estudo 2	63
5.3	Módulo M1 com o condicionamento entre a válvula <i>bypass</i> e os controladores real e virtual	65
5.4	Módulo M2 com o condicionamento relativo ao sinal do motor da válvula de água	66
5.5	Módulo M3 com o condicionamento relativo ao sensor de caudal	67
5.6	Primeiro módulo desenvolvido para interface do sensor de temperatura NTC	67
5.7	Módulo M4 com o condicionamento relativo ao sinal do sensor de temperatura	68
5.8	Módulo 5 - Circuito de interface para o sinal PAM de acionamento do ventilador	70
5.9	Módulo 6 - Circuito de interface para o sinal do sensor de Hall do ventilador	70
5.10	Exemplo de comutação para 2 sinais distintos	71
5.11	Exemplo de comutação do sinal PWM da válvula <i>bypass</i>	73
5.12	Receção do sinal PWM vindo do microcontrolador	73
5.13	Envio do sinal PWM para o componente real	73
5.14	Possível fluxo experimental para o caso de estudo 2	74

Lista de Acrónimos

HILS Hardware-In-The-Loop-Simulation

RTW Real-Time Workshop

PCI Peripheral Component Interconnect

ISA Industry Standard Architecture

TCP-IP Transmission Control Protocol - Internet Protocol

UUV Unmanned underwater vehicle

RTWT Real Time Windows Target

PWM Pulse-Width Modulation

RTI Real Time Interface

Bosch TT Bosch Termotecnologia

Capítulo 1

Introdução

1.1 Enquadramento

Nos dias de hoje, a população procura cada vez mais novas tecnologias, com maior eficiência e com a melhor qualidade possível, de forma a melhorar o seu conforto. Isso leva a que a indústria produtiva esteja sob pressão constante para o desenvolvimento de novos produtos, com o objetivo de satisfazer as necessidades do mercado. Além da pressão do consumidor, há também a pressão da própria indústria que tem a necessidade constante de se impor à concorrência para conseguir ganhar os mercados e obter maiores lucros. Tudo isso obriga as empresas a encurtar o tempo para desenvolvimento e testes de novos equipamentos.

Para os ensaios de equipamentos é necessário definir novas estratégias que permitam efetuar testes numa fase o mais inicial possível do projeto, poupando tempo e despesas de fabrico. A utilização de modelos dinâmicos tem vindo a ganhar terreno no desenvolvimento de novas estratégias de controlo, pela facilidade com que se torna possível otimizar os controladores reais em diferentes modos de operação dos equipamentos, incluindo os que dificilmente ocorrem na prática. Assim, torna-se possível poupar algum tempo e despesas prevendo o comportamento de um determinado sistema sem risco de experimentação. A este tipo de teste denomina-se de *Hardware-In-The-Loop-Simulation (HILS)*.

Apesar de todas as tentativas de melhoria e rapidez no processo de desenvolvimento, são necessárias plataformas de ensaio capazes de se adaptarem a diferentes formas de teste. As bancadas de ensaio têm um papel fundamental nesta componente, servindo como apoio ao desenvolvimento e permitindo testar equipamentos ou controladores para detetar erros de concepção e evitar maiores prejuízos.

A Bosch Termotecnologia (Bosch TT) é um dos exemplos de rigor e empenho, destacando-se pela sua capacidade de inovação e desenvolvimento, elevando o aquecimento de águas domésticas a outro nível. Assim, graças à implementação contínua de novas tecnologias e novos produtos, a marca tornou-se líder mundial no que concerne a sistemas térmicos de aquecimento de águas domésticas. Esta empresa, tal como as outras, necessita de grande investimento na área de testes de forma a facilitar e acelerar o processo de desenvolvimento de novos conceitos e de novos produtos.

Nesta dissertação, em parceria com a Bosch TT, foram abordadas algumas das necessidades e as maiores dificuldades no processo de testes de controladores e equipamentos, com o objetivo de gerar um conceito que possa facilitar o processo de ensaios.

1.2 Apresentação da problemática

Na Bosch TT, tal como em qualquer empresa de desenvolvimento, existe uma grande necessidade de testar os novos controladores e equipamentos na sua fase de conceção. A empresa caracteriza-se pela sua elevada capacidade de resposta às necessidades do mercado no que diz respeito à produção de esquentadores e termo-acumuladores. Para isso, os engenheiros precisam de desenvolver novos produtos, de forma mais eficiente e rápida, que se traduz numa redução de tempo de desenvolvimento e na necessidade de novas ferramentas de teste.

As bancadas de ensaio têm um papel fundamental no apoio aos investigadores neste processo para testar os novos equipamentos e controladores, de forma a garantir o adequado funcionamento dos mesmos. No entanto, estas apresentam algumas limitações, destacando-se o facto de apenas possibilitarem o teste de *hardware* numa fase mais avançada do projeto. Na maioria dos casos, para testar um controlador é necessário desenvolver todo o equipamento e programar o microcontrolador para otimizar o funcionamento do mesmo. Todo este processo de geração de conceito, desenvolvimento, construção, programação e otimização demora o seu tempo e obriga a que haja uma grande colaboração entre os vários setores de desenvolvimento de uma empresa.

Proposto pelos colaboradores da Bosch TT, foi mostrado o interesse em criar uma bancada de ensaios que fosse mais flexível que as existentes, e pudesse ser mais útil numa fase ainda mais inicial do projeto.

Na verdade, a problemática apresentada não está só presente na Bosch e pode alargar-se a várias empresas de desenvolvimento, de setores completamente distintos. Uma bancada de testes mais flexível e inovadora poderia ser uma grande ajuda na maioria da indústria de desenvolvimento.

1.3 Objetivos

Com a presente dissertação pretende-se desenvolver e implementar uma metodologia para teste de estratégias de controlo em sistemas de aquecimento de águas domésticas. Para isso é necessário estudar as diferentes possibilidades de *hardware* e software que permitam *hardware-in-the-loop-simulation*, e identificar as principais limitações das bancadas existentes.

De seguida pretende-se propor um conceito de bancada que permita não só aplicar a técnica HILS, mas também se torne flexível ao ponto de permitir outras formas de teste e simulação que possam ser úteis no processo de desenvolvimento e teste de controladores.

Destacam-se os seguintes objetivos:

- Recorrer a modelos dinâmicos em MATLAB/Simulink de sistemas de aquecimento de águas, e adaptá-los para os testes necessários;
- Implementação de ferramentas de software e *hardware* para utilizar metodologias de simulação com *hardware-in-the-loop* para teste de controladores;
- Construção de uma bancada de ensaios que permita realizar diferentes tipos de testes, de forma rápida e eficaz.

Para o efeito deve-se elaborar um caso de estudo que envolva aquecimento de água, de forma a comprovar o bom funcionamento do conceito de bancada proposto. Para tal é necessário aplicar, nesse caso de estudo, as etapas principais do desenvolvimento de um controlador. Por fim, pretende-se efetuar um segundo caso de estudo aplicado a um

esquentador da Bosch TT, com o objetivo de reportar os passos essenciais para aplicar as técnicas abordadas anteriormente.

1.4 Guia de leitura

Nesta secção aborda-se a forma como está organizada a presente dissertação e os respetivos conteúdos de cada capítulo.

Em primeiro lugar faz-se uma breve revisão bibliográfica e tecnológica onde se aborda a temática do *hardware-in-the-loop-simulation* e o respetivo *hardware* e software que permite aplicar a técnica. Cada um é exemplificado com um trabalho já realizado, fazendo uma análise comparativa entre eles. Ainda se procuram trabalhos efetuados em bancadas de ensaios, ou que envolvam experimentações, de forma a apurar algumas das características existentes para servir de apoio ao presente trabalho.

De seguida, no Capítulo 3, com base nas pesquisas efetuadas e nas necessidades enumeradas pelos colaboradores da Bosch TT, apresenta-se o conceito de bancada de ensaios proposto abordando as principais características e vantagens da mesma.

No Capítulo 4 aborda-se um caso de estudo, que envolve todas as possíveis etapas de projeto e desenvolvimento de um controlador, com a aplicação das técnicas pretendidas e mostrando o conceito de bancada de ensaios, aproveitando as suas vantagens, com o objetivo de comprovar o bom seu funcionamento.

Após comprovar o funcionamento do conceito de bancada, no Capítulo 5, inicia-se um segundo caso de estudo, através do projeto dos módulos de condicionamento elétrico e com as indicações principais para o seu avanço.

Por último, no Capítulo 6, apresentam-se as conclusões do trabalho, com o balanço do trabalho efetuado e com as principais ilações retiradas de ambos os casos de estudo. Abordam-se também algumas propostas de trabalhos futuros.

Capítulo 2

Revisão do Estado da Arte

Nesta secção apresenta-se uma breve revisão do estado da arte do tema subjacente a este trabalho, nomeadamente no que diz respeito a *Hardware-in-the-loop-simulation* e bancadas de testes. Sendo o *software* e o *hardware* das partes mais importantes e que maior investimento requerem, abordam-se de seguida as várias opções de *software* e *hardware*, e alguns dos exemplos de aplicação. Por último, efetua-se uma análise comparativa dos mesmos. Ainda se faz uma breve revisão tecnológica a nível de bancadas de ensaios, mostrando-se algumas das suas características.

2.1 Hardware in the loop simulation

O HILS é uma técnica com forte utilidade para desenvolvimento e teste de sistemas embebidos. Esta ganha ainda maior relevância quando esses sistemas não podem ser testados com facilidade no seu ambiente operacional. Requer o desenvolvimento de uma simulação em tempo real, contemplando algumas, ou todas as partes do sistema real. [1]

É uma técnica de simulação que mistura tanto elementos virtuais como reais, permitindo assim aproximar a experiência a um cenário mais real. Podemos ver na Figura 2.1 as possibilidades de interação dos diferentes componentes, reais e virtuais, num sistema de controlo comum. Esta característica torna o HILS bastante útil para avaliação e desenvolvimento de controladores, reduzindo o risco de experimentação. Com isto, torna-se não só possível a poupança de investimento e de tempo, ao evitar consequências de erros de projeto inicial dos controladores, permitindo a identificação e correção dos mesmos de forma mais prática e em tempo útil [2].

Apesar destas serem algumas das fortes razões da utilização do HILS, há outras vantagens que têm tornado esta técnica cada vez mais usual na indústria do desenvolvimento, onde se destacam as seguintes [3]:

- Prototipagem rápida: Ainda que se necessite de algum tempo para modelar os sistemas e criar o simulador, é um processo que na maior parte dos casos acaba por ser mais rápido que construir totalmente o *hardware*.
- Se alguma das partes do sistema virtual for difícil de representar, esta pode ser facilmente substituída pelo componente real.
- É uma técnica que permite simulações de fenómenos físicos complexos de forma mais rápida que as puramente virtuais. Por exemplo, um circuito de controlo real pode ser facilmente testado em diferentes controladores simulados, ou até condições de ambiente

diversificadas podem ser simuladas com simples alterações de parâmetros.

- Tem a possibilidade de repetir um mesmo processo de forma semelhante várias vezes, o que dificilmente é possível em protótipos inteiramente físicos com grande variação de ambientes

- Possibilita a simulação de eventos destrutivos sem risco. Por vezes existe a necessidade de testar o comportamento de um sistema em casos extremos, por exemplo a própria destruição do sistema, e com o HILS é possível fazê-lo de uma forma segura e sem despesas de componentes.

- Permite que diferentes equipas possam trabalhar ao mesmo tempo em diferentes partes do *hardware* de um sistema, sem os problemas de integração deste.

Uma condição imprescindível para a realização de HILS é a capacidade de simulação e resposta, em tempo real, dos modelos matemáticos dos sistemas.

O hardware-in-the-loop pode ser aplicado desde sistemas relativamente simples aos mais complexos. As primeiras utilizações foram na indústria militar, no estudo do comportamento e desenvolvimento de mísseis, na indústria espacial, nos sistemas de controlo de aeronaves, e na indústria nuclear, nos controladores de reatores nucleares [1].

As aplicações mais conhecidas do HILS referem-se, provavelmente, aos simuladores de voo, onde inicialmente se pretendia representar os componentes de um cockpit e mais tarde para simular uma cabine e a experiência de voo na totalidade para a formação de pilotos. Neste último os componentes eram reais, apenas se simulando o comportamento da aeronave e os seus movimentos [4].

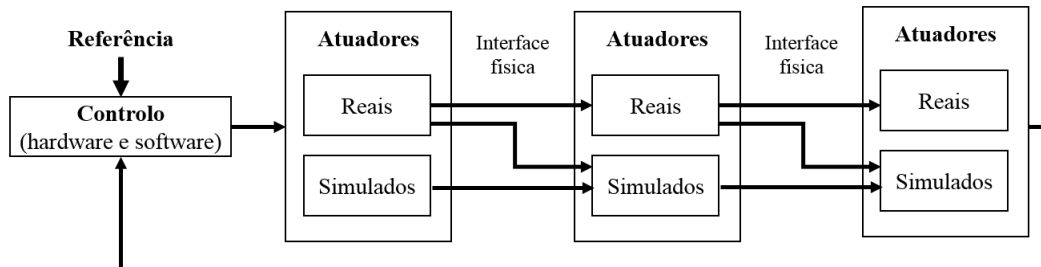


Figura 2.1: Possibilidades de interação numa experiência HILS em sistemas de controlo (Adaptado de [5])

2.2 Revisão tecnológica para HILS

De forma a concretizar a simulação HIL com sucesso é necessária a utilização de *software* e *hardware* apropriado para criar os modelos dinâmicos a serem simulados, para gerar o código de execução correspondente, criar a interface do sistema e correr a simulação em tempo real. Com o aumento das necessidades e das aplicações do *hardware-in-the-loop* houve um acompanhamento do mercado, com a oferta de *software* e *hardware* apropriado.

De seguida abordam-se algumas das ferramentas utilizadas para HILS, e exemplos de aplicação das mesmas.

2.2.1 *MATLAB/Simulink*

O *MATLAB/Simulink* é o *software* mais referenciado no que diz respeito a modelação de sistemas dinâmicos. Desenvolvido pela MathWorks, o *MATLAB* permite uma grande capacidade de cálculos numéricos, que a par do *Simulink* possibilita o desenvolvimento e simulação de modelos dinâmicos de sistemas complexos. Os sistemas podem ser modelados a alto nível através das interações entre as funções dos blocos do *Simulink*, enquanto que o *MATLAB* executa os cálculos desses blocos [6]. Importa referir que a generalidade do *hardware* para simulação em tempo real permite interação com o *MATLAB/Simulink*.

2.2.2 Real-Time Workshop

O *Real-Time Workshop (RTW)* é uma extensão das capacidades do *MATLAB/Simulink*. Este permite gerar automaticamente o código executável a partir dos modelos do *Simulink* para aplicar em simulações em tempo real. Além disso tem uma interface gráfica simples e intuitiva e uma arquitetura aberta e extensiva [7]. Em [8] explorou-se uma adaptação de um controlo *skyhook* para um sistema de suspensão Marpherson semi-ativa de braço hidráulico. É implementado um novo modelo orientado ao controlo que incorpora o movimento de rotação da massa não suspensa. No que concerne à componente da simulação, foi utilizada uma suspensão variável, como *hardware* real, uma unidade de controlo, uma unidade de atuação por solenóide, uma placa da dSPACE e um computador para correr a simulação em tempo real. O algoritmo de controlo foi desenvolvido através do *MATLAB/Simulink* e, com o auxílio do RTW, gerou-se o código em linguagem C desse algoritmo a partir do seu diagrama de blocos. Esse código foi carregado na placa dSPACE através do Real-Time-Interface, permitindo correr a simulação em tempo real. Além do exemplo anterior, são muitas as aplicações HILS que recorrem ao RTW para gerar o código automaticamente a partir do *MATLAB/Simulink*, o que demonstra a sua utilidade e indispensabilidade na simulação em tempo real.

2.2.3 xPC Target

O *xPC Target* é um exemplo de um sistema operativo de tempo real. Desenvolvido pela MathWorks, é uma das soluções de *software* para prototipagem, teste e desenvolvimento de sistemas em tempo real. Permite conectar os modelos *Simulink* a modelos físicos e executá-los num computador com *hardware* compatível. Possibilita execução de *hardware-in-the-loop* e processamento de dados, adicionando blocos I/O aos modelos. Requer o RTW, que gera automaticamente o código e faz o seu *download* para um segundo computador que corre o xPC Target *real-time kernel*. Através do carregamento e execução automática do *kernel* em tempo real a aplicação possibilita que o computador *target* (computador que corre simulação em tempo real) e o *host* (computador que contém o modelo dinâmico) trabalhem de forma independente um do outro [9].

Placas de aquisição de dados como as *Peripheral Component Interconnect (PCI)* e as *Industry Standard Architecture (ISA)* são compatíveis e recomendadas para este *software*. As placas PCI são preferíveis às ISA na aquisição de sinais em tempo real, pois permitem *plug-and-play*, recursos de interrupção, comunicação mais ampla e rápida e *software* de calibração. [10]

Uma aplicação do *xPC Target* foi efetuada em [11]. Pretendeu-se comprovar a utilidade e benefícios do HILS numa fase preliminar do desenvolvimento de sistemas em

manipuladores robóticos. A arquitetura para este trabalho dividiu-se em interface do utilizador, computador para simulação, *hardware* e sistema de controlo. Um computador é responsável pela interface e monitorização, contendo o *MATLAB/Simulink* para construir os modelos para a simulação e para o controlo. O *Real-Time Workshop* gera, a partir dos blocos *Simulink*, o executável para a simulação em tempo real que são enviados, por comunicação *Transmission Control Protocol - Internet Protocol (TCP-IP)*, para o computador responsável pelo controlo e simulação, respetivamente. Ambos os computadores contêm o *xPC Target Real-Time Kernel*, de forma a possibilitar a simulação em tempo real, o que não seria permitido com um sistema operativo como o Microsoft Windows, por exemplo.

2.2.4 VxWorks

Produzido pela Windriver Systems, o VxWorks [12] é um sistema operativo de tempo real. Executa o código em intervalos de tempo específicos, garantindo os recursos computacionais necessários.

O VxWorks é utilizado para sistemas embebidos em tempo real, onde cada processo tem o seu próprio espaço de memória, que contém o executável do programa e os respetivos dados. É possível que vários processos estejam em memória uma só vez e que cada um possa conter múltiplas tarefas.

Não se encontram muitos exemplos de aplicação deste *software*, destacando-se o trabalho em [13]. O objetivo deste trabalho passou pela implementação de um algoritmo de controlo de um *Unmanned underwater vehicle (UUV)* em ambiente *MATLAB/Simulink*, a sua conversão para um código executável e a simulação do mesmo em ambiente real com técnicas *hardware-in-the-loop*.

A modelação e simulação do sistema em análise é complexa, exigindo grande capacidade computacional. Tal como já foi referido, para modelação deste sistema recorreu-se ao *Simulink*, através do estudo de um modelo dinâmico e algoritmo de controle de um UUV já desenvolvido.

No que concerne à geração dos códigos, recorreu-se ao *Real Time Workshop* para gerar um código que possibilitasse interação com o *Constellation framework*. O *Constellation framework* tem capacidade para interface e geração do código executável, partindo desde o modelo desenvolvido em *MATLAB/Simulink* até à criação da linguagem de programação ANSI C++. Este possibilita a conversão do modelo num único bloco simples, composto de entradas e saídas.

Após todos esse processo de tratamento utilizou-se o VxWorks para permitir a simulação em tempo real.

2.2.5 Real-Time Windows Target

O *Real Time Windows Target (RTWT)* é uma opção para prototipagem e teste de sistemas em tempo real, fornecida pela MathWorks. É utilizado num computador em conjunto com o *MATLAB/Simulink* para criar modelos dinâmicos.

Este *software* usa um *real-time kernel* para assegurar que a aplicação é executada em tempo real. O *real-time kernel* corre num *desktop* e interceta a interrupção do relógio do computador antes do sistema operativo Windows recebê-lo, bloqueando assim todas as chamadas para o mesmo. O *kernel* de seguida usa essa interrupção para disparar a

execução do modelo compilado, conseguindo dar prioridade máxima à aplicação. Para criar o executável recorre-se a um compilador, tipicamente o *Real-Time Workshop*, que através do modelo *Simulink* cria um código em linguagem C ou noutro tipo de linguagem preferível.

A comunicação é feita com o *hardware* através de uma placa I/O de aquisição de dados; ao executar os modelos em tempo real, o RTWT capta os dados das entradas dos canais e usa-os como entradas para o modelo de diagrama de blocos, da mesma maneira que os processa e envia de volta para o sistema em simulação. Uma das vantagens deste *software* é a compatibilidade com as placas de aquisição de dados, contendo os *drivers* para um grande número de placas de aquisição diferentes [14].

Este *software* possibilita que um computador funcione como *host* e como *target* numa simulação em tempo real [15].

Observe-se o exemplo prático de aplicação deste *software* em [16]. O "*Mechatronic Aeropendulum*" baseia-se num pequeno motor DC ligado a uma das extremidades de uma haste. As unidades motoras, alimentadas por uma placa elétrica, fazem a hélice rodar e permitem a vara balançar. A posição angular é medida por um potenciômetro ligado na zona de articulação, na outra extremidade da haste. Foi criado um módulo de *MATLAB/Simulink*, para ler o ângulo do pêndulo e enviar um sinal de comando para o motor, que corre no *Real Time Windows Target*, permitindo uma taxa de amostragem de 200 Hz. No que concerne à simulação, recorreu ao RTWT que foi primeiramente utilizado para compilar o modelo *Simulink* e gerar o código C ou C++. De seguida, o *kernel* permitiu que o projeto corresse em tempo real partilhando dados com o microcontrolador.

2.2.6 RT-LAB

O RT-LAB é uma plataforma de simulação em tempo real que permite a simulação de modelos, aquisição de dados e controlo. Consegue converter os modelos *Simulink* rapidamente para simulação em tempo real, sendo útil principalmente para aplicações HILS [17]. Esta plataforma permite gerar o código automaticamente através do *Simulink*, facilitando a vida aos engenheiros, que apenas precisam de arrastar os blocos para o modelo gráfico e conectar as respetivas entradas e saídas, sem qualquer preocupação com a programação de baixo nível [18].

É possível ver várias aplicações do RT-LAB em [18]. O objetivo deste trabalho passou pela apresentação de uma nova ferramenta para simulação em tempo real e controlo de sistemas eletrónicos. Essa ferramenta é um *software* que simula circuitos elétricos, ARTEMIS, e uma plataforma de tempo real, o RT-LAB, para correr os modelos *Simulink* no computador. Após a construção do modelo no *Simulink*, um bloco do ARTEMIS foi adicionado para executar uma simulação *offline*, ou seja, no ambiente *Simulink*. De seguida esse modelo foi aberto pelo RT-LAB e compilado, sendo necessária uma conversão para código através do *Real-Time Workshop*. Os autores destacam 3 aplicações: Num transistor conversor AC-DC, num conversor *Pulse-Width Modulation (PWM)* DC-AC, e simulação em tempo real de conversores de comutação de alta tensão.

2.2.7 *software* dSPACE

A dSPACE disponibiliza *software* e *hardware* para prototipagem rápida e para o HILS. A modelação dos sistemas dinâmicos pode ser desenvolvida no *MATLAB/Simulink* enquanto que a componente de aquisição de dados e controlo e geração de código é da

responsabilidade da dSPACE. Esta fornece um conjunto de cartas de aquisição de dados orientadas para a simulação em tempo real de sistemas e para implementação de controladores, garantindo a execução dos algoritmos de controlo em tempo real. O *Real Time Interface (RTI)* é o elo de ligação entre o *MATLAB/Simulink* e o *hardware*. O RTI tem a capacidade de gerar o código para simulação de forma muito simples e sem grande trabalho para o utilizador. Enquanto que o RTW gera o código do modelo, o RTI fornece os blocos que implementam as capacidades I/O dos sistemas da dSPACE nos modelos do *Simulink*, preparando-o para a aplicação em tempo real. O modelo em tempo real é compilado, carregado e iniciado automaticamente no *hardware* de tempo real sem a necessidade de escrever código intermediário [19].

Pode-se observar como um exemplo o trabalho [5], cujo objetivo passou por propor uma metodologia para a organização de bibliotecas de modelos, um conjunto de modelos híbridos semi-empíricos para componentes de sistemas hidráulicos, e uma plataforma para a realização de experiências de simulação com *hardware-in-the-loop*. No que à componente experimental diz respeito, foi projetado e desenvolvido um manipulador de três eixos de atuação hidráulica. A nível de *hardware* e *software* para a aquisição de dados recorreu uma carta da National Instruments (PCI1200), mais recomendada para a aquisição de dados embora se possa aplicar em controlo. O sistema de modelação utilizado foi o *Simulink* e o núcleo de tempo real o *Real Time Windows Target*, da Mathworks. A par da PCI1200 ainda recorreu à DS1102 da dSPACE, esta mais dirigida à simulação em tempo real de sistemas e implementação de controladores. O código de tempo real, gerado em C, foi feito através do *Real Time Workshop* e do *Real Time Interface*. A interação com o programa foi realizada através da aplicação ControlDesk.

2.2.8 National Instruments - PCI

As cartas de aquisição de dados PCI, criadas pela National Instruments, apresentam algumas vantagens e desvantagens. É possível destacar uma grande variedade de seleção de entradas e saídas disponíveis, velocidade de comunicação disponível e possibilidade de operações *plug-and-play*. O facto de apenas ser possível o seu uso em *desktops* apresenta-se como uma desvantagem. Das cartas PCI existentes, e após alguma comparação de conteúdos, é possível destacar a PCI-6259. Esta carta de aquisição pode trabalhar em sistemas operativos como o Linux, MAC ou Windows. Tem 16 entradas analógicas e 4 saídas analógicas, ambas com resolução de 16 bits. Também tem 48 E/S digitais com canais bidirecionais.

2.2.9 Análise comparativa entre *software* e *hardware*

Todas as possibilidades apresentadas anteriormente apresentam vantagens e desvantagens claras. Havendo a possibilidade de escolher *software* e *hardware*, o primeiro deve ser prioridade pois é este que vai definir o tipo de *hardware* a utilizar. São várias as combinações possíveis conforme se pode ver na Figura 2.2

A nível de *software* para criação dos modelos dinâmicos o recomendado é o *MATLAB/Simulink*. É o mais usual, e tem compatibilidade com a maior parte do *software* de *hardware-in-the-loop*.

Conforme se referiu anteriormente, requiere um compilador de forma a gerar o código executável a partir do *Simulink*. Se a opção for pela dSpace tem-se o RTI, como alternativa existe o RTW.

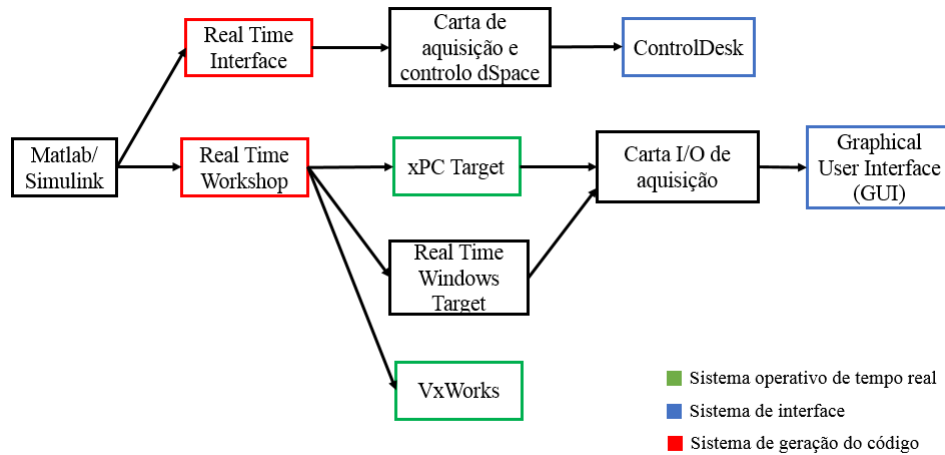


Figura 2.2: Possibilidades de interação entre *software* e *hardware*

No que diz respeito a *software* de tempo real há três possibilidades. Destaca-se a possibilidade de optar por um sistema operativo em tempo real, um simulador que pode correr num sistema operativo orientado a eventos, e usar *hardware* próprio que possa encarregar-se da simulação. Na primeira opção, o mais comum é o xPC Target e o VxWorks, não havendo grande diferença aparente entre eles; no entanto, sendo o xPC Target criado pela mesma empresa do *MATLAB/Simulink* poderá ser uma mais valia. Uma das desvantagens de ambos é que necessitam de um outro computador para correr a aplicação em tempo real. O RTWT é útil na medida em que pode ser usado no mesmo computador onde se faz a modelação do sistema. Tendo em conta o avanço computacional nos dias de hoje, acredita-se que isso não será entrave na capacidade de simulação. Este pode correr em Windows, o que é uma mais valia e assim utilizar-se apenas um computador para modelação, simulação e interface.

No caso da dSpace, a situação é diferente visto que fornece cartas de aquisição de dados que além de possuírem I/O, para aquisição e envio de sinais, também têm a capacidade de efetuar controlo e correrem as aplicações em tempo real. Isto apresenta-se como uma vantagem pois apenas se necessita de um computador, visto que este leva uma menor carga computacional ao correr a simulação na carta. A interface pode ser facilmente gerada através do ControlDesk, que além de intuitivo é também *user friendly*. A empresa oferece várias soluções de *hardware* com grande capacidade a nível de aquisição e envio de sinais, quer analógicos, quer digitais, e de comunicação série.

Analisando as opções, aquelas que parecem cumprir da melhor forma com os requisitos mínimos são a Dspace e o Real Time Windows Target visto que em ambas só se necessita de um computador. Na primeira é necessário uma carta de aquisição e controlo que corra a simulação, enquanto no segundo uma simples carta I/O é suficiente. Observe-se a Tabela 2.1, com o resumo do *hardware* recomendado para cada um dos casos.

Tabela 2.1: Tabela resumo de *hardware* e *software* para simulação em tempo real

<i>software</i>	Hardware	Entradas Analógicas	Saídas Analógicas	Entradas digitais	Saídas digitais	Saídas PWM
RTI + ControlDesk	DS1104,R&D Controller Board	8	8	20	20	10
	DS1007,PPC Processor Board					
	DS2202,HIL I/O Board	16	20	38	16	9
Real Time Windows Target + RTW	MF624 - PCI multifunction I/O	8	8	8	8	4
	NI PCI-6713	0	8	8		2
	NI PCI-6014	16	2	8		2
VxWorks + RTW	NI PCI-6713	0	8	8		2
	NI PCI-6014	16	2	8		2
xPC Target + RTW	NI PCI-6713	0	8	8		2
	NI PCI-6014	16	2	8		2
RTLAB	NI PCI-6713	0	8	8		2
	NI PCI-6023E	16	0	8	8	2

2.3 Bancadas de ensaio

Conforme já foi referido, um dos objetivos do trabalho passava por desenvolver um conceito de bancada de ensaios que possa realizar, de entre várias opções, testes HILS. Além disso pretende-se desenvolver um conceito inovador que possibilite várias formas de teste. Importa dessa forma sustentar esse estudo para saber que necessidades a nível de *software* e *hardware* uma bancada normal exigiria, orientando o estudo para sistemas térmicos.

Apresentam-se de seguida alguns exemplos de trabalhos efetuados relacionados com bancadas de ensaios com maior ênfase em sistemas térmicos, enunciando as principais características de cada um.

2.3.1 Bancada para estudo de comportamento de esquentador

No trabalho [20] efetuou-se um estudo do comportamento de esquentadores de várias marcas, de forma a traçar curvas de funcionamento para comparar modos de funcionamento e rendimentos. O objetivo era obter dados relativos à temperatura da água à entrada e à saída do esquentador, à temperatura dos gases resultantes da combustão à saída do permutador de calor e também a sua composição, e os caudais de gás combustível e água.

Para efetuar esses testes foi necessário desenvolver uma bancada experimental com os requisitos mínimos de instrumentação e segurança. Dos sensores para instrumentação destacam-se os medidores infravermelhos e paramagnéticos, termopares, sensores de caudal e transdutores de pressão. A bancada foi ainda equipada com um sistema de aquisição, ao qual foi associado o respetivo condicionamento de sinal para leitura dos sensores.

No que diz respeito à interface e monitorização dos ensaios desenvolveu-se um programa em Basic. Este foi responsável pela interpretação dos sinais analógicos e digitais da carta de aquisição de dados, bem como pela geração dos ficheiros com os dados de medição para tratamento. Através da aplicação era possível ainda enviar o sinal de abertura da eletroválvula de água de entrada para iniciar os ensaios.

2.3.2 Projeto de bancada para a Vulcano

Torna-se difícil abordar sistemas térmicos e não falar da Bosch TT. No âmbito de um estágio curricular desenvolveu-se o trabalho [21] onde se pretendeu efetuar um projeto de renovação total dos sistemas de testes da Vulcano Termodomésticos, atual Bosch TT.

As bancadas existentes nos laboratórios da Vulcano, na altura, eram equipadas com pouca tecnologia e recursos a nível de instrumentação. O trabalho realizado pretendeu desenvolver um sistema integrado para aquisição e análise de dados relativos a ensaios efetuados a esquentadores. Para a sua execução inicialmente foram identificados os principais requisitos da bancada de ensaios comum e elaborado o plano de melhorias da mesma a nível de *software* e *hardware*, de forma a tornar as bancadas existentes tecnologicamente mais evoluídas.

A bancada desenvolvida permitiu um sistema robusto de aquisição de dados através da solução *FieldPoint* da National Instruments. O módulo implementado corre uma aplicação em tempo real usando o *software* Labview Real-Time para interface e monitorização. Além disso, foram adicionados mais transdutores de temperatura e pressão, comparando com as bancadas anteriores. Outra das vantagens é a possibilidade de efetuar os testes sem a necessidade de ter o computador ligado, usando mostradores independentes.

Com este trabalho as bancas passaram a ter maior segurança e um *software* robusto, moderno, com maior capacidade de aquisição de sinais, de gerar ficheiros com informação e tudo isto com uma interface *user-friendly* para o utilizador, aumentando a eficiência dos ensaios.

2.3.3 Simulação de um sistema de arrefecimento de um motor

É também importante observar o trabalho efetuado noutro tipo de sistemas, não só os térmicos de aquecimento mas também de arrefecimento. [22] É um trabalho que pretende comparar os resultados de comportamento de um sistema de arrefecimento de um motor em ambiente simulado e em ambiente experimental, através de uma bancada de ensaios.

No que diz respeito à componente de simulação, foi modelado o sistema no *software* AMESim, através das equações de balanço de energia inerentes ao processo. O AMESim é um *software* com bibliotecas próprias que permite implementar equações matemáticas para traduzir o comportamento dos sistemas.

Ao nível da bancada de ensaios, esta foi construída para testar a rejeição de calor no sistema. A instrumentação envolveu essencialmente dois sensores de temperatura e um de caudal. Esta apenas serviu de teste e aquisição de dados através de *hardware* e *software* próprio.

Para efetuar os testes os autores definiram uma temperatura de *Setup* e testaram primeiro no sistema virtual e de seguida no sistema real, adquirindo os dados para um ficheiro para posterior comparação.

2.3.4 Desenvolvimento de bancada para sistema de refrigeração

No trabalho [23] construiu-se uma bancada de ensaios de raiz, através de processos mecânicos de soldadura e programação, para testar modos de falha em sistemas de refrigeração.

Esta bancada foi equipada com controlador, circuito de refrigeração, material de instrumentação e aquisição de dados. O sistema funciona em modo manual onde, através do uso de um botão de seleção, se escolhe o modo de falha que se pretende simular.

Para controlo do modo de falha que se pretende foi desenvolvido um programa em linguagem Ladder, para receber o sinal do botão de escolha do modo de falha e ativar o relé correspondente ao mesmo.

O sistema de monitorização implementado mostra, de forma gráfica, o modo de falha que está ativo e o que está a acontecer, através da amostragem em gráficos das temperatura ao longo do tempo.

2.3.5 Controlador de velocidade de motores recorrendo ao *MATLAB/Simulink* e *dSPACE*

Foi encontrado um trabalho interessante relativo ao controlo de velocidade de motores elétricos [24]. Este projeto, apesar de não incluir sistemas térmicos, é de destacar na medida em que envolve toda a componente de simulação, modelação e teste em tempo real em sistemas físicos.

O objetivo desta tese passou por efetuar testes aos controladores de velocidade de motores elétricos, realizada por conversores eletrónicos de potência, usando a simulação em tempo real.

O primeiro passo envolveu a modelação do sistema físico em *MATLAB/Simulink* através da implementação das equações dinâmicas do sistema. De seguida, com esse modelo, simulou-se o comportamento do mesmo e retiraram-se os valores.

Depois do processo virtual, iniciou-se a preparação do programa para a simulação em tempo real através dos blocos da biblioteca *dSPACE*. Todo o processo de aquisição de sinais foi efetuado numa bancada própria equipada com computador e com uma carta de aquisição de dados, a DS1104 Controller Board em conjunto com o CP1104 I/O Board. Para monitorização do sistema e tratamento de dados recorreu-se ao ControlDesk.

A componente interessante deste trabalho é o facto de ter realizado testes em controladores virtualmente e em seguida experimentalmente, em tempo real, de forma a verificar os seus comportamentos e abrindo a possibilidade de otimizar os mesmos no sistema físico.

2.3.6 Teste a uma bomba de água de um sistema térmico

Numa dissertação para a Universidade de British Columbia, [25] efetuou-se um estudo à performance de uma bomba de água de um sistema térmico através de vários ensaios experimentais e análise de dados.

Para os ensaios o autor montou uma bancada experimental com todo o equipamento necessário para teste e aquisição de dados, e também com vários sensores e atuadores inerentes ao sistema físico. De forma a efetuar a instrumentação adquiriu os dados dos sensores de temperatura, de caudal e de potência, e fez a aquisição de dados através de uma carta de aquisição da National Instruments. A interface do utilizador para visualização e registo dos dados foi efetuada a partir do LabVIEW.

De seguida foi modelado o sistema num *software* próprio. Através dos dados experimentais do teste parametrizou-se o modelo virtual de forma a aproximar os resultados. Para isso, em ambos os testes usou várias temperaturas de *setpoint* e compararam-se as diferenças, fazendo assim as respetivas aproximações. Este sistema modelado possibilitou efetuar experiências em condições limite, o que não seria possível testar no equipamento existente.

Este foi mais um exemplo de um trabalho aplicado a um sistema térmico com testes numa bancada de ensaios a um modelo real, que permite apenas teste e aquisição de dados e a modelação de um sistema virtual onde, usando os dados experimentais, se parametriza o modelo.

2.3.7 Balanço acerca das bancadas de ensaios

Apesar de apenas se referirem alguns exemplos de bancadas de ensaios as pesquisas efetuadas foram alargadas. Com base nessas pesquisas, foi possível retirar várias ilações e pensar uma alternativa para colmatar algumas das limitações identificadas.

Todo e qualquer processo de desenvolvimento de produto, seja mecânico, eletrónico ou térmico, envolve construção de protótipos que necessitam de ser testados para se identificarem eventuais problemas e implementar as melhorias necessárias.

Conforme se pôde verificar a maior parte das bancadas de ensaios tem a instalação do sistema a ser testado, com os respetivos sensores para instrumentação, e tem o *software* de aquisição e interface, de forma a visualizar o estado das variáveis e as mudanças ao longo do tempo. Estas basicamente permitem o teste dos sistemas físicos e a sua monitorização.

Algumas aplicações, como é o caso do trabalho referido na Secção 2.3.5, permitem que os sistemas físicos sejam diretamente ligados ao modelo dinâmico e controlados por este através de *software* de tempo real e *hardware* de aquisição de dados. Desta forma é possível, além da monitorização do sistema, testar os controladores diretamente no sistema real. Esta é uma prática que tem alguma utilização em alguns tipos de sistemas, contudo muito difícil de encontrar em sistemas térmicos.

De uma forma resumida, encontraram-se três tipos de bancada: as que permitem testar o modelo real e o respetivo controlador e monitorizar o comportamento; as que permitem testar o modelo real usando um controlador simulado e efetuar a monitorização; e as que permitem manualmente criar algumas instabilidades e verificar como respondem os protótipos.

Após as pesquisas efetuadas, e analisando os problemas referidos na Secção 1.2, no capítulo seguinte, vai-se projetar um conceito de bancada que consiga superar algumas das limitações referenciadas e se adapte a mais formas de testes.

Capítulo 3

Conceito de bancada de ensaios

O primeiro passo para avançar de forma estruturada e concisa passou pelo estudo de uma proposta de *layout*. Desta forma, com base nas pesquisas efetuadas, propôs-se uma bancada flexível e adaptável a diferentes fases do projeto de sistemas e controladores da Bosch TT.

Pretende-se mostrar de seguida algumas das funcionalidades da bancada e as características que a destaca das existentes. Assim apresenta-se a proposta de *layout* e a escolha de *hardware* e *software* necessários para a sua realização.

3.1 Proposta de conceito de *layout*

Um dos principais objetivos desta dissertação passa por elaborar um conceito de bancada que seja inovador e possibilite a realização de testes HILS. No entanto, em discussão com os representantes das Bosch TT foram mencionadas outras necessidades no projeto de sistemas que seria muito importante se fossem colmatadas. Assim, estudou-se um *layout* de bancada que cumpra com os objetivos iniciais da dissertação, vá ao encontro das necessidades dos engenheiros da Bosch TT, e que desempenhe mais algumas funções que podem ser de grande utilidade no projeto e desenvolvimento de sistemas, seja no ramo de sistemas térmicos ou outro.

Analisaram-se as bancadas disponibilizadas e usadas para elaborar trabalhos de instrumentação e controlo onde se identificaram algumas limitações, conforme se verificou na Secção 2.3.7. Com base nessas limitações e nas necessidades expressas pelos colaboradores da Bosch foi pensado um conceito de bancada que consiga resolver a maior parte desses problemas.

Verificou-se que, com base na pesquisa efetuada, a maior parte das bancadas de ensaios apenas permite realizar testes em sistemas reais, havendo algumas exceções em que possibilitam também testes de controladores em modelos virtuais. No entanto, se a pesquisa for restringida a sistemas térmicos as opções de teste e simulação são ainda mais reduzidas.

Com base em todas as limitações detetadas, e com o objetivo de implementar um conceito de bancada diferente do que já existe, elaborou-se o esquema de bancada da Figura 3.1. Esta é constituída pela parte real do sistema, por exemplo um esquentador, pelo controlador real, tipicamente um microcontrolador, e por último pelo sistema de simulação, monitorização e aquisição de dados, constituído pelo computador que contém

o *software* próprio para modelação dos sistemas e simulação dos mesmos e também pelo *hardware* de aquisição e controlo. Importa referir que todos estes sistemas necessitam de ser alimentados, e estar interligados através de módulos de condicionamento de sinal que ajuste os níveis de tensão de cada componente.

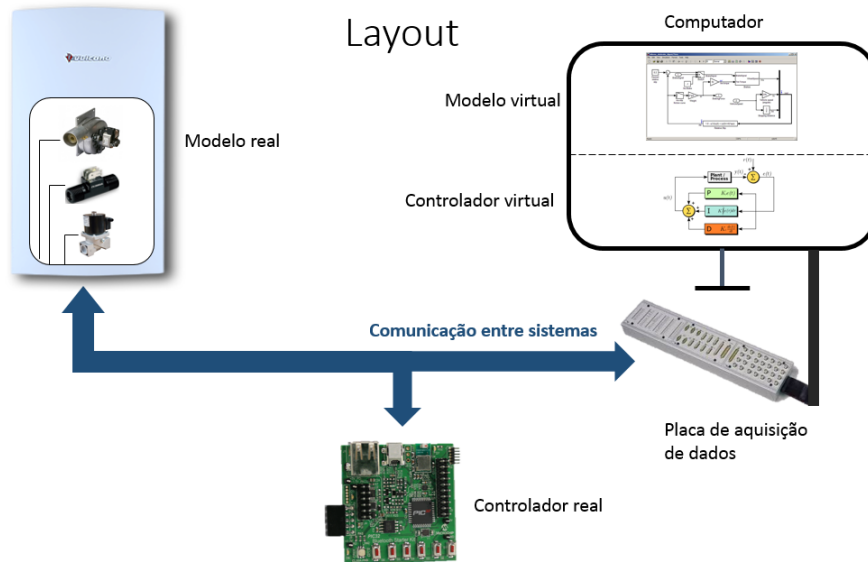


Figura 3.1: Esquema da proposta de layout esquemática da bancada de ensaios

3.2 Funcionalidades do conceito

Torna-se necessário explicitar em que medida é que o conceito apresentado é inovador e quais as funcionalidades que o distinguem. Sabe-se que o projeto normal de um controlador se inicia com a modelação quer do sistema quer do controlador em ambiente simulado, a que se segue a implementação desse controlador num dispositivo de controlo real e, por último, pelos testes num protótipo físico do sistema.

Uma das grandes desvantagens da passagem do controlador virtual para o real é o facto de apenas se saber o seu comportamento num sistema modelado e, só após a programação do microcontrolador, é que se torna possível testá-lo num sistema real. Existe neste processo alguma perda de tempo e recursos que poderiam ser evitados caso se pudesse saber o comportamento do controlador, e a otimização do mesmo, no sistema real ainda antes da programação do microcontrolador.

Com o objetivo de facilitar o projeto de sistemas térmicos e criar um conceito inovador, pensou-se numa bancada que oferecesse mais funcionalidades de teste, conforme se descreve de seguida:

Modo 1 - Uma das primeiras opções que a banca permite é efetuar a simulação sem qualquer componente real incluído, onde se tem o modelo dinâmico do sistema, o controlador virtual projetado e um sistema de tempo real que permita correr a simulação. Na Figura 3.2 é possível ver o referido módulo esquematizado, onde se mostra o controlo virtual no sistema virtual.

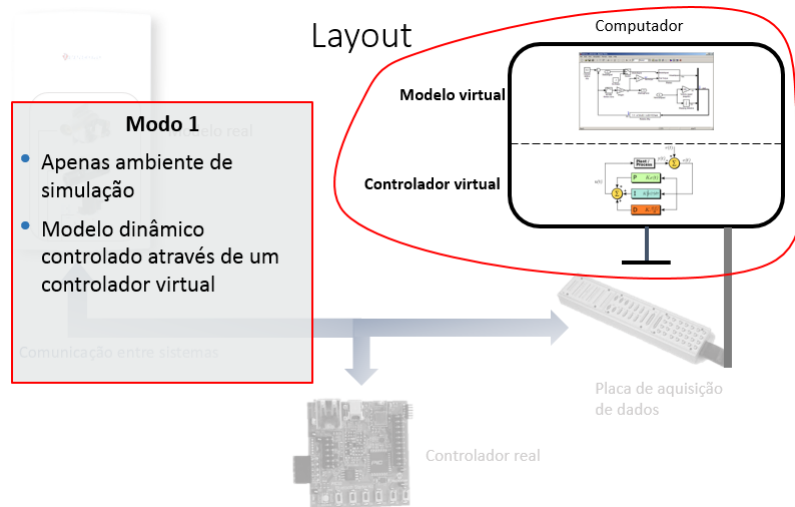


Figura 3.2: Modo 1 de funcionamento da bancada de ensaios

Modo 2 - Outra das funções desta bancada possibilita que o controlador projetado em Simulink controle e monitorize diretamente o sistema real, seja ele um esquentador ou outro aparelho (Figura 3.3). Esta possibilidade apresenta-se como uma vantagem visto que é possível, numa fase inicial do desenvolvimento, testar os controladores e otimizá-los já com o comportamento real dos sistemas. Assim, é possível verificar e melhorar os parâmetros de controlo virtual de forma a que sejam implementados mais tarde no microcontrolador com maior grau de confiança. Este modo representa o controlo virtual do sistema real.

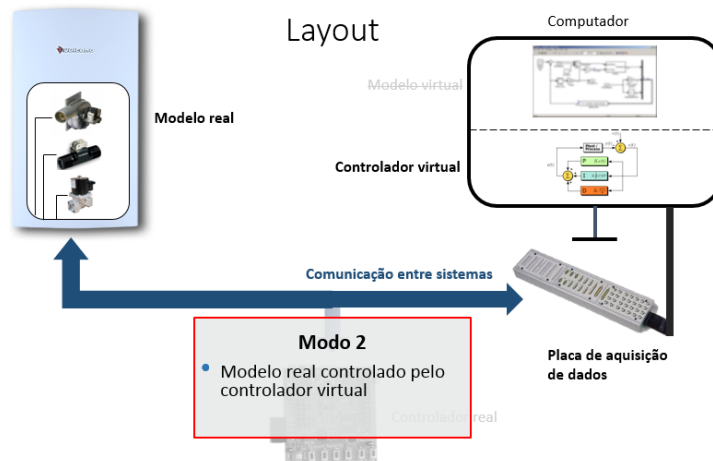


Figura 3.3: Modo 2 de funcionamento da bancada de ensaios

Modo 3 - Como terceira opção existe a possibilidade de efetuar o denominado *hardware-in-the-loop-simulation*, ou seja, o controlo real do sistema virtual (Figura 3.4). Neste apenas se tem como componente real o controlador, que comunica com o computador pela carta de aquisição de dados e onde os componentes do sistema são modelados. O modelo virtual é então controlado pelo microcontrolador, como se de um sistema real se tratasse.

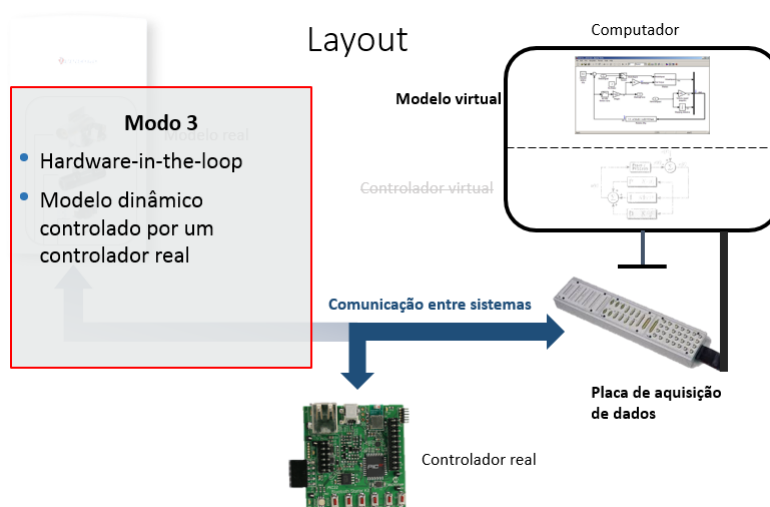


Figura 3.4: Modo 3 de funcionamento da bancada de ensaios

Modo 4 - Por último, é possível testar o controlador real no sistema real (Figura 3.5). Esta funcionalidade pode ser útil numa fase final do projeto, em que todos os componentes são reais. Esta é uma função que já é comum na maior parte das bancadas de ensaios de testes existentes nas diversas áreas.

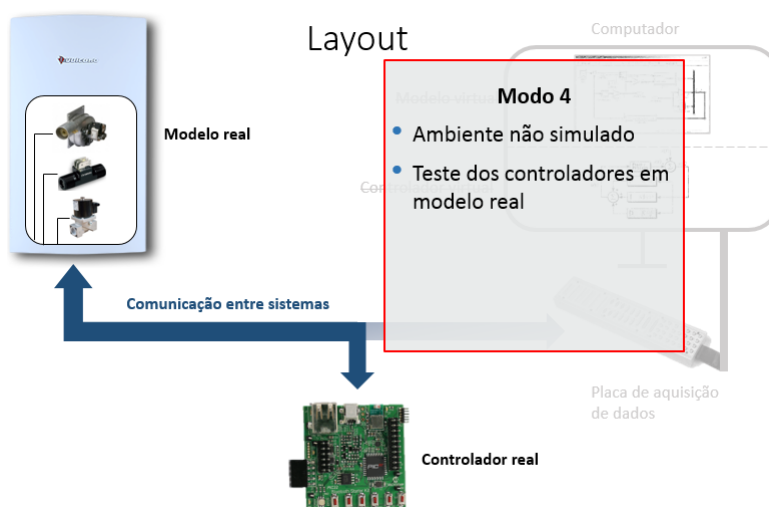


Figura 3.5: Modo 4 de funcionamento da bancada de ensaios

Sem grandes alterações, é possível testarem-se diferentes componentes, de diversas formas, com maior ou menor parcela de simulação incluída. A conjugação destas quatro funcionalidades torna esta bancada única, não só pela sua flexibilidade como também pelos potenciais problemas que resolve durante o desenvolvimento de um novo sistema.

Com base no conceito mencionado, e nas funções que a bancada pode desempenhar, efetuou-se um modelo ilustrativo 3D da banca, recorrendo ao *software* CATIA V5, para auxiliar na perceção do espaço que cada componente pode ocupar, conforme se pode ver na Figura 3.6. Esta já representa as potenciais ligações de água e gás, um esquentador

como sistema a testar, um microcontrolador, as placas de aquisição de dados, computador e quadro de alimentação.



Figura 3.6: Proposta de layout em 3D da bancada de ensaios

3.3 Características técnicas

3.3.1 Escolha de *hardware* e *software* para simulação e monitorização

Conforme já foi revisto no Capítulo 2, é necessário um conjunto de *hardware* e *software* que permita, entre várias opções, simulação, aquisição e simulação com *hardware-in-the-loop*.

Deve-se escolher uma carta de aquisição de dados, responsável pela aquisição e envio dos sinais e, em alguns casos, por correr a aplicação em tempo real.

No que diz respeito ao *software* de tempo real, deve-se ter em atenção à compatibilidade com a carta de aquisição de dados e com o *software* de modelação. Este oferece bibliotecas que interagem com *hardware* e *software*, de forma a permitir correr a aplicação em tempo real e fornecer bibliotecas ao *software* de modelação para trabalhar com sinais que vêm do exterior. É também necessário orientar a instalação de ambos para interagirem um com o outro. Tipicamente, em primeiro o *software* de tempo real e só de seguida o de modelação, como é o caso do Release 4.2 da dSPACE e o *MATLAB* 7.0.4.

A escolha do computador é também um dos parâmetros a ter em conta, sendo necessário um sistema operativo compatível com o *software* de tempo real e de modelação. Além disso, é importante ter memória RAM suficiente para correr aplicações mais pesadas e encurtar o tempo para as carregar e simular.

3.3.2 Condicionamento de sinal

O condicionamento de sinal é indispensável para o funcionamento deste conceito de bancada de ensaios. Há uma série de sinais, todos diferentes entre si, que estarão interligados e necessitam da adaptação necessária para não haver danificação de *hardware*. A ideia para esta bancada é desenvolver os módulos necessários para cada componente, de forma a adaptar os sinais para poderem ser ligados ao microcontrolador e à carta de aquisição de dados.

Para os sinais existentes entre o microcontrolador e a carta de aquisição de dados os módulos são simples de obter, visto que os sinais digitais podem ser diretamente ligados, e apenas poderá ser necessária uma adaptação para os sinais analógicos. Ainda assim, os últimos podem ser conectados diretamente caso se considere a gama de funcionamento de 0 a 5 VDC.

No que diz respeito à ligação entre os componentes reais e o microcontrolador ou a carta de aquisição de dados, é preciso um cuidado maior devido aos sinais de tensão. Por exemplo, supondo uma válvula de gás on/off cuja tensão de funcionamento é de 24 VDC, existem duas formas de a controlar, uma através do microcontrolador e outra através do *hardware* de simulação. Então, seriam enviados os sinais digitais tanto de um como de outro que, antes de chegarem à válvula, teriam de ser adaptados por um módulo que transformasse essa tensão em 24 VDC, o que atuaria a válvula sem problemas.

3.3.3 Comutação entre modos de operação

Pretende-se uma bancada de ensaios que possibilite várias formas de teste para auxiliar toda a fase de desenvolvimento do produto. Além disso, também é importante que a mudança entre o tipo de simulação que a bancada oferece seja simples e rápida.

Essa comutação pode ser feita de várias formas, ou manualmente ou virtualmente, sendo que é importante haver contactos para permitir a ligação entre pares de sistemas para efetuar determinado teste.

A Figura 3.7 representa um esquema do que foi abordado. Basicamente tem-se o exemplo de dois componentes com os respetivos módulos "M" de condicionamento de sinal, que são desde logo adaptados para uma tensão de funcionamento geral de forma a ligar à carta de aquisição de dados e ao controlador real. Apesar de não estar representado, pode haver a necessidade de também haver módulos entre o controlador real e a carta de aquisição, dependendo, como é claro, dos requisitos de ambos.

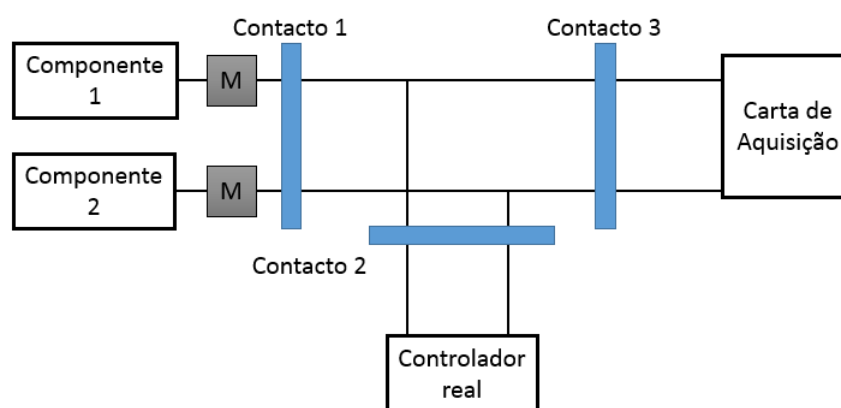


Figura 3.7: Esquema representativo com módulos de condicionamento e contactos

Para mostrar melhor o funcionamento deste conceito, e as suas utilidades práticas, será de seguida apresentado um caso de estudo que envolva as questões que foram abordadas, e que comprovam que o conceito é válido através de várias experiências práticas num sistema térmico.

Capítulo 4

Caso de estudo 1

Após a proposta de *layout* e funcionalidades da bancada de ensaios apresentadas, torna-se necessário proceder a testes que possam sustentar que é possível aplicar estratégias de controlo numa só bancada de quatro formas diferentes. Esses tipos de testes, tal como já foi referido anteriormente, passam pelo controlo virtual no sistema virtual, controlo virtual no sistema real, controlo real no sistema virtual e controlo real no sistema real.

De forma a comprovar que é possível implementar as quatro funcionalidades de teste a uma bancada de ensaios recorreu-se a um caso de estudo simples. Explica-se de seguida esse caso de estudo com os principais passos a nível da modelação do sistema, programação do controlador e condicionamento de sinal, e demonstra-se recorrendo a testes práticos que o conceito proposto é passível de ser aplicado.

4.1 Definição do caso de estudo 1

Para demonstrar que as funcionalidades de bancada de ensaios apresentada no Capítulo 3 são possíveis de implementar, procurou-se um caso de estudo que as consiga desempenhar. O caso de estudo desenvolvido está relacionado com o aquecimento de água, não só pela sua aproximação a sistemas de aquecimento de águas domésticas como também pelo reduzido número de aplicações encontradas de *hardware-in-the-loop* em sistemas térmicos.

O caso de estudo elaborado está esquematizado na Figura 4.1. Este é constituído por um reservatório aberto com um tubo para entrada de água e outro para saída no fundo. A entrada de água é controlada por uma bomba de água "B" e a saída por uma válvula manual "V". O interior é constituído por uma resistência elétrica de aquecimento de água "R" e um sensor de nível "N". A temperatura é medida à entrada e no interior do reservatório por dois sensores de temperatura "T1" e "T2".

Através dos componentes escolhidos, e para elaborar o plano de modelação e controlo, definiu-se o princípio de funcionamento do sistema. Assim, o objetivo é escolher uma temperatura de *setup* e tentar que a resistência de aquecimento aqueça a água de forma a manter a temperatura interior próxima desse valor. À saída tem-se uma válvula manual que permite regular o caudal de saída, fazendo com que a coluna de água desça. De forma a tentar manter essa coluna de água constante tem-se uma bomba de água à entrada, que atua de forma on-off cada vez que a coluna de água não for detetada pelo sensor de nível. Assim, tem-se entrada de água à temperatura ambiente, saída de água e aquecimento para a fazer chegar a uma temperatura definida pelo utilizador.

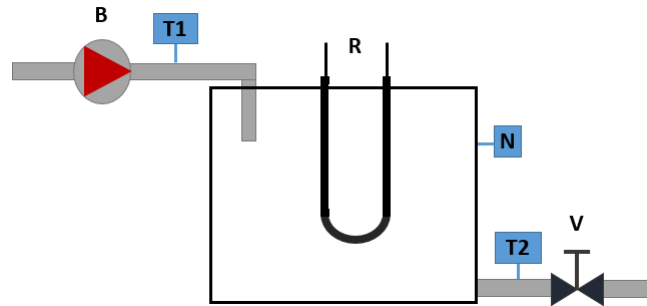


Figura 4.1: Representação de um controlador com *feedback* e *feedforward*

Apesar da simplicidade do conceito, o importante neste caso de estudo é provar que o conceito da bancada de ensaios proposto é válido e funciona, desde que se preparem os modelos dinâmicos de forma correta e se projetem os módulos de interface de sinal.

É objetivo desta experiência demonstrar o funcionamento das várias possibilidades de teste da bancada, modelando o sistema e o controlador em *MATLAB/Simulink* efetuando simulações, e de seguida experimentá-lo no sistema real, depois programar o microcontrolador com os mesmos parâmetros do projetado virtualmente e testá-lo no sistema virtual. Por último, controlar o sistema real através do microcontrolador e comparar todos os testes. Deseja-se que todos os testes sejam alternados com a maior rapidez e com a menor quantidade de adaptações possível.

4.1.1 Descrição da bancada

Já foi abordado no capítulo anterior o conceito da bancada de ensaios e a forma como se pretende que opere. No entanto, apesar de também se detalhar nas secções seguintes, é importante ter uma noção do funcionamento desta bancada de ensaios.

Para concretizar este caso de estudo pretende-se um reservatório de água, conforme já referido, equipado com os sensores e atuadores necessários para controlo de temperatura, aquecimento e entrada e saída de água. Este é o sistema real que será colocado em teste de forma a ajudar na validação do conceito.

Para controlo do sistema é necessário um controlador real, neste caso um *Arduino Uno*. Este é responsável por controlar o sistema real no modo de operação 4 e o sistema virtual no modo de operação 3.

O sistema virtual é constituído por *hardware* e *software*, computador com o *MATLAB* e *Simulink* para modelação dos sistemas, *software* da *dSPACE* para interface e monitorização e, finalmente, para adquirir os sinais e efetuar o controlo, correndo as aplicações em tempo real, tem-se a carta de aquisição e controlo DS1104.

Os três sistemas referidos anteriormente têm que estar interligados através de ligações elétricas. No entanto, alguns deles funcionam com sinais diferentes, pelo que é necessário desenvolver módulos de condicionamento de sinal, que serão abordados posteriormente.

Para ligar os pares de sistemas, alternando entre os diferentes modos de operação, existirá um comutador responsável por atuar os contactos existentes junto a cada secção, conforme o tipo de teste que se pretenda.

A Figura 4.2 esquematiza um primeiro *layout* da bancada, onde é possível observar como será a sua organização.

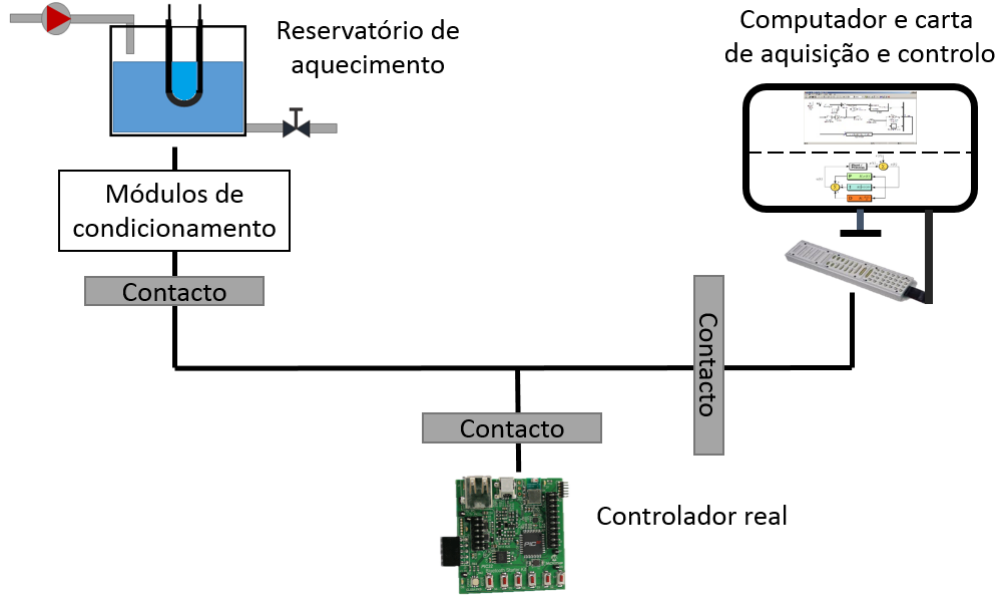


Figura 4.2: Esquema com o *layout* representativo do caso de estudo 1

Os passos seguintes passam por obter a modelação do sistema referido e o respetivo controlador, o controlador real, e o sistema real.

4.2 Modelação do Caso de Estudo 1

De forma a implementar o sistema representado pela Figura 4.1 em *Matlab/Simulink*, é necessário considerar as equações que descrevem o seu comportamento.

Os principais parâmetros a ter em conta são a potência da resistência de aquecimento, o caudal de entrada, o caudal de saída e a transferência de calor no fluido e para o exterior.

A equação de conservação da massa para um sistema aberto estabelece que:

$$\dot{m}_{in} + \dot{m}_{gen} - \dot{m}_{out} = \dot{m}_{sto} \quad (4.1)$$

Como no interior de um volume de controlo não temos geração de massa, a taxa de acumulação pode ser expressa pelo caudal de entrada e de saída. Neste caso, a passagem de caudal mássico para caudal volúmico é direta dada a massa volúmica da água ser constante (Equação 4.2):

$$\dot{V}_{sto} = \dot{V}_{in} - \dot{V}_{out} \quad (4.2)$$

Desenvolvendo fica-se com:

$$A_{res} \frac{\partial h}{\partial t} = \dot{V}_{in} - \dot{V}_{out} \quad (4.3)$$

Em que h é a altura da água no reservatório e A_{res} a área (constante) da sua secção transversal. Por último, obtemos uma relação do nível da água em função dos caudais de entrada e saída:

$$h = \int \frac{\dot{V}_{in} - \dot{V}_{out}}{A_{res}} dt \quad (4.4)$$

Através da Equação 4.2 consegue-se uma relação entre a altura do líquido no reservatório e os caudais de entrada e de saída.

Para o caso de estudo apresentado considera-se que o caudal de saída do reservatório apenas vai depender da diferença de pressões no orifício de saída. A equação para determinar o caudal, através de um orifício, sob a ação da força gravítica, é dada pelo Teorema de Torricelli:

$$\dot{V}_{out} = C_d S \sqrt{2gh} \quad (4.5)$$

No que diz respeito à transferência de calor no sistema sabe-se que a equação de conservação de energia de um sistema aberto estabelece que:

$$\frac{\partial E_{cv}}{\partial t} = \dot{Q} + \dot{W} + \sum_{in} \dot{m}e + \sum_{out} \dot{m}e \quad (4.6)$$

Aplicando as simplificações devidas, chega-se à Equação 4.7.

$$\frac{\partial E_{cv}}{\partial t} = \dot{Q} + \rho_{agua} c_{agua} (T_{in} \dot{V}_{in} - T_{out} \dot{V}_{out}) \quad (4.7)$$

Através do desenvolvimento da variação da energia com o tempo fica-se com:

$$\frac{\partial E_{cv}}{\partial t} = \rho_{agua} V c_{agua} \frac{\partial T_{out}}{\partial t} \quad (4.8)$$

Com a simplificação e os cálculos intermédios, chega-se a uma equação diferencial final que depende dos caudais e das temperaturas de entrada e saída, da potência calorífica aplicada ao sistema e da altura da coluna de água. Esta equação estabelece a variação temporal da temperatura da água como:

$$\frac{\partial T_{out}}{\partial t} = \frac{1}{\rho_{agua} A_{res} h c_{agua}} \left[\dot{Q} + \rho_{agua} c_{agua} (T_{in} \dot{V}_{in} - T_{out} \dot{V}_{out}) \right] \quad (4.9)$$

A equação anterior refere-se a um sistema sem perdas pelas paredes do reservatório. Falta agora adicionar a componente de transferência de calor por condução, pelas paredes do reservatório, e de convecção, pela superfície da água. A Equação 4.10 apresenta a potência conduzida pelas paredes do reservatório, onde se considerou apenas os valores por condução.

$$\dot{Q}_{cond} = \frac{T_{s1} - T_{s2}}{\frac{L}{k_t A}} \quad (4.10)$$

No caso da Equação 4.10, L representa a espessura das paredes do reservatório, k_t é a condutividade térmica do material e A é a área da superfície de contacto. A variável T_{S1} corresponde à temperatura do lado interior da parede do reservatório e T_{S2} é a temperatura do lado exterior dessa parede. Para simplificação assumiu-se que T_{S1} é a temperatura da água e T_{S2} a temperatura do ar.

A Equação 4.11 mostra a convecção existente na parte superior do reservatório pela superfície da água.

$$\dot{Q}_{conv} = h_{conv} A (T_s - T_{ar}) \quad (4.11)$$

Na Equação 4.11, h_{conv} é o coeficiente de convecção, A a área da superfície, T_s é a temperatura à superfície da água e T_{ar} a temperatura do ar envolvente.

O problema de transferência de calor devido às perdas pode ser descrito pelo esquema da Figura 4.3, que demonstra as variáveis que foram referidas anteriormente e que estão diretamente relacionadas com esse processo.

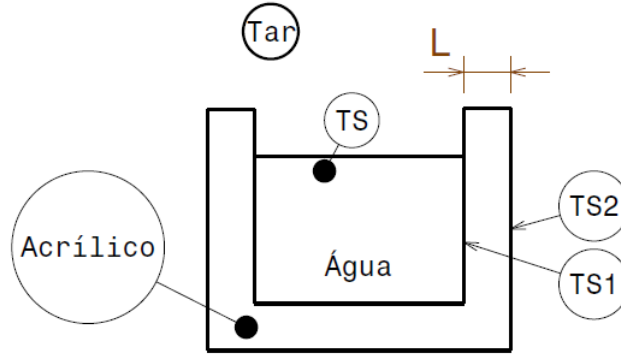


Figura 4.3: Representação das variáveis envolvente nas perdas por condução e convecção

Assim, associando as expressões das perdas à equação de conservação de energia e efetuando as simplificações devidas, obtém-se a seguinte expressão para a variação da temperatura da água:

$$\frac{\partial T_{out}}{\partial t} = \frac{1}{\rho_{agua} A_{res} h c_{agua}} \left[\dot{Q}_{aquecimento} - 5 \times \frac{T_{out} - T_{ar}}{\frac{L}{k_t A_{res}}} - h_{conv} A_{res} (T_s - T_{ar}) + \rho_{agua} c_{agua} (T_{in} \dot{V}_{in} - T_{out} \dot{V}_{out}) \right] \quad (4.12)$$

Considerou-se que a potência térmica adicionada ao sistema corresponde à potência da resistência de aquecimento ($\dot{Q}_{aquecimento}$). No caso do caudal de entrada considerou-se que é o máximo dado pela bomba quando está ligada e zero quando esta está desligada, ficando assim modelados os principais parâmetros que descrevem o sistema térmico referido.

4.3 Controlador virtual

Após a implementação dos modelos em *MATLAB/Simulink* iniciou-se o processo de controlo do mesmo. Foi escolhido um controlador sem grande preocupação em pormenorizar, nesta fase, a sua eficiência, apenas melhorado após os testes no sistema real.

Houve o cuidado de aplicar um controlador que fosse normalmente implementado em sistemas térmicos, que é o controlo *feedforward*. Na maior parte dos controladores existe uma referência, ou *setpoint*, que é pretendida à saída do sistema. Através de, por exemplo, um PID calcula-se o erro entre a saída e a referência, e o controlador responde de forma a eliminar esse erro, denominado *feedback control*. No caso do *feedforward*, calcula-se uma componente teórica de potência inicial e, após o cálculo do *feedback control*, adiciona-se à resposta do controlador essa parcela teórica. É possível observar na Figura 4.4 um diagrama com a representação dos dois tipos de controlador referidos.

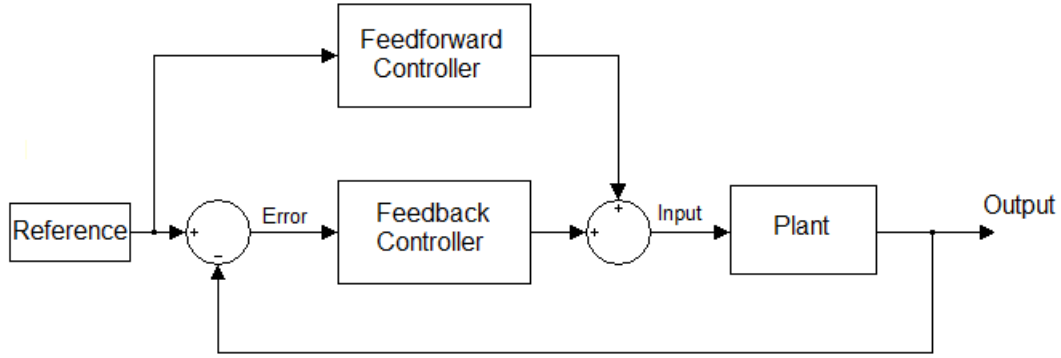


Figura 4.4: Representação de um controlador com *feedback* e *feedforward* (Adaptado de [26])

Importa referir que o controlo está a ser efetuado com base na potência térmica a ser entregue ao sistema. Em primeiro lugar calculou-se a potência teórica que o sistema deve ter usando a temperatura desejada e a do ambiente. Assim, desenvolveu-se a Equação 4.9, chegando à seguinte simplificação:

$$P_{Teorica} = \rho C_{agua} (\dot{V}_{out} T_{des} - \dot{V}_{in} T_{in}) + 5 \times \dot{Q}_{cond} + \dot{Q}_{conv} \quad (4.13)$$

Ao mesmo tempo é calculada a potência efetiva (Equação 4.14), ou seja a potência necessária em cada instante com os valores de temperatura reais:

$$P_{Efetiva} = \rho C_{agua} (\dot{V}_{out} T_{out} - \dot{V}_{in} T_{in}) + 5 \times \dot{Q}_{cond} + \dot{Q}_{conv} \quad (4.14)$$

Os valores anteriores subtraem-se com o objetivo de calcular um erro de potência:

$$Erro = P_{Teorica} - P_{Efetiva} \quad (4.15)$$

De seguida esse erro é multiplicado por cada uma das componentes do controlador PID e obtém-se assim:

$$PID = Erro Proporcional + Erro Integral + Erro Derivativo \quad (4.16)$$

O resultado da Equação 4.16 aplicado ao sistema resultaria no denominado *feedback control*, tal como referido anteriormente. Para obter o *feedforward control* basta, para chegar à potência final, adicionar a componente teórica do sistema.

$$P_{Final} = P_{Teorica} + PID \quad (4.17)$$

Nesta fase, não houve grande preocupação no projeto do controlador, tendo em conta que não esse o é objetivo desta dissertação. Ainda assim, na Secção 4.12, recorre-se a uma das funcionalidades da bancada para encontrar o melhor controlador.

4.4 Comutação entre modos de utilização da bancada

Um dos objetivos deste trabalho prende-se com a facilidade com que se poder alterar o tipo de teste em causa, com o mínimo de alterações possível no que diz respeito às mudanças dos sinais. A comutação entre os modos de operação da bancada é feita através de um comutador de quatro posições:

- A - Modo 4 (Ligação do sistema real com arduino);
- B - Modo 3 (Ligação entre arduino e carta de aquisição);
- C - Modo 2 (Ligação entre carta de aquisição e sistema real);
- D - Modo 1 (Correspondente à simulação em ambiente unicamente modelado e virtual).

Esse comutador será responsável por comutar três relés principais que controlam as ligações entre cada um dos sistemas referidos. Pode-se observar na Figura 4.5 o esquema de comutação, onde está representado o layout da bancada com o sistema real, a carta de aquisição de dados, o controlador e os relés 1, 2 e 3 que controlam a ligação entre os sistemas. Pretende-se que cada posição do comutador ative um par de relés, para gerar a ligação, e que essa combinação seja controlada por outros relés intermédios. Por exemplo, se o operador pretender acionar o modo 2 de funcionamento, ou seja, controlar o modelo real através do controlador simulado, necessita de acionar os relés 1 e 3.

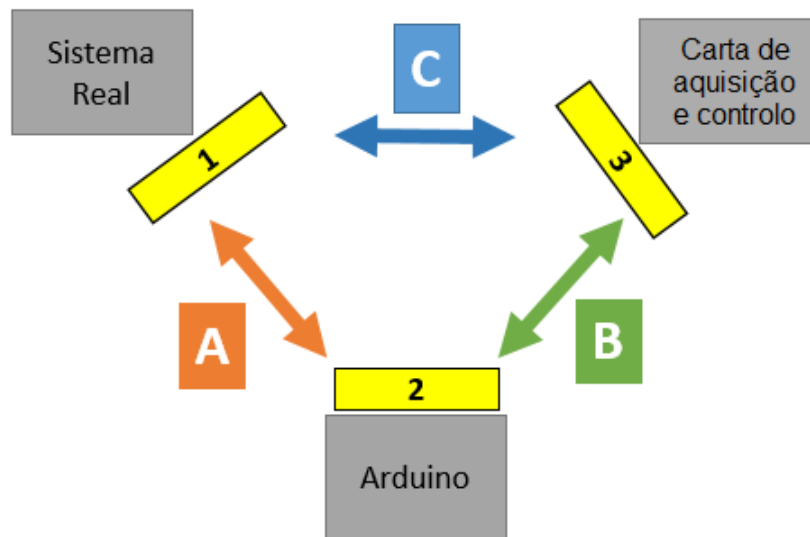


Figura 4.5: Esquema de comutação entre simulações

Para determinar quantos relés diferentes seriam necessários para gerar todas as combinações possíveis recorreu-se a uma tabela de verdade (Tabela 4.1). Para efetuar essa tabela é preciso saber que para a opção A tem que se fechar os contactos 1 e 2, para a B os contactos 2 e 3, para a C os contactos 1 e 3 e para a D todos os contactos têm que estar abertos.

Tabela 4.1: Tabela de verdade para os contactos de comutação

Posições do botão				Relés		
A	B	C	D	1	2	3
0	0	0	0	x	x	x
0	0	0	1	0	0	0
0	0	1	0	1	0	1
0	0	1	1	x	x	x
0	1	0	0	0	1	1
0	1	0	1	x	x	x
0	1	1	0	x	x	x
0	1	1	1	x	x	x
1	0	0	0	1	1	0
1	0	0	1	x	x	x
1	0	1	0	x	x	x
1	0	1	1	x	x	x
1	1	0	0	x	x	x
1	1	0	1	x	x	x
1	1	1	0	x	x	x
1	1	1	1	x	x	x

De seguida, recorreu-se a mapas de Karnaugh para cada um dos relés 1, 2 e 3. O objetivo é simplificar todas as possibilidades de ligação e determinar quais os contactos mínimos necessários para o seu estabelecimento. Este procedimento está representado nas Tabelas 4.2, 4.3 e 4.4, respetivamente.

Tabela 4.2: Mapa de Karnaugh da saída 1

		CD			
		00	01	11	10
AB	00	x	0	x	1
	01	0	x	x	x
	11	x	x	x	x
	10	1	x	x	x

Tabela 4.3: Mapa de Karnaugh da saída 2

		CD			
		00	01	11	10
AB	00	x	0	x	0
	01	1	x	x	x
	11	x	x	x	x
	10	1	x	x	x

Tabela 4.4: Mapa de Karnaugh da saída 3

		CD			
		00	01	11	10
AB	00	x	0	x	1
	01	1	x	x	x
	11	x	x	x	x
	10	0	x	x	x

Simplificou-se cada um dos Mapas de Karnaugh e obteve-se para cada saída as seguintes equações lógicas correspondentes aos contactos 1, 2 e 3:

$$1 = C + A \quad (4.18)$$

$$2 = A + B \quad (4.19)$$

$$3 = C + B \quad (4.20)$$

Assim, para atuar os relés de saída 1, 2 e 3 usaram-se outros três relés com dois contactos cada, atuados por cada posição A, B, C e D do botão rotativo, de forma a gerar todas as combinações necessárias para se ter as várias formas de ligação entre sistemas.

Importa referir que, além dos relés anteriores, seriam necessários mais 3 relés junto à carta de aquisição de dados, de 2 contactos cada, para permitir as trocas entre as saídas e as entradas analógicas e as entradas e saídas digitais sem a necessidade de trocar os cabos de posição. Por exemplo, supondo que a temperatura do reservatório estaria a ser adquirida na entrada analógica 1, para controlo da temperatura do reservatório, e que logo de seguida se pretendia simular o sistema virtual a ser controlado pelo arduino, o que implicaria enviar o valor da temperatura simulada pela saída analógica 1. Assim, tem-se 2 canais diferentes que trabalham com a temperatura do reservatório e, sem os relés mencionados, seria necessário trocar a posição dos cabos. Neste caso de estudo, por falta de material, faz-se a troca de posição do cabo, no entanto, no Apêndice A apresenta-se o projeto elétrico adequado para uma maior simplicidade e rapidez de comutação.

4.5 Condicionamento do sinal

Para a interação entre os vários sistemas deste caso de estudo é preciso desenvolver o condicionamento de sinal, que permita que o sistema real, o arduino e a carta de aquisição de dados estejam interligados. Existe uma grande variedade de sinais, conforme se pode verificar na Tabela 4.5, pelo que importa efetuar o projeto elétrico detalhado para evitar erros de aquisição de material e de construção dos circuitos. Os módulos de condicionamento de sinal que serão projetados referem-se apenas às ligações entre o sistema real e os sistemas de controlo. No que diz respeito às ligações entre o controlador real e o sistema virtual vão-se enviar os sinais na mesma gama de funcionamento de forma a poupar em componentes de interface.

Tabela 4.5: Gama de sinais de alimentação de saída/entrada

Componente	Tensão de saída/entrada	
Resistência elétrica	230 VAC	
Bomba de água	230 VAC	
Sensor de nível	24 VDC	
Sensor de temperatura	200 mV/°C	
Arduino	Alimentação	9 VDC
	Entradas lógicas	0-5 VDC
	E/S Digitais	5 VDC
DS1104 R&D Controller Board	E/S Analógicas	0-10 VDC
	E/S Digitais	5 VDC

4.5.1 Drive de potência

A bomba de água e a resistência, que funcionam a 230 VAC, são acionados através de sinais digitais tanto da carta de aquisição como do arduino a 5 VDC. É necessário garantir o isolamento e a capacidade de comutação necessária entre os 5 VDC e os 230 VAC e para isso optou-se por relés de estado sólido. Estes têm um diodo emissor de luz que fecha o contacto da alimentação dos 230 VAC, garantindo assim o isolamento necessário. No caso do acionamento da resistência elétrica, como esta é de elevada potência, tanto o relé de estado sólido (SHIMADEN PAC03B-010) como os cabos de ligação são mais robustos. Tendo em conta que a corrente dos sinais digitais é baixa em demasia para acionar os relés, recorreu-se a um transistor NPN e a um diodo, conforme se pode verificar no circuito representado na Figura 4.6, para proteção das saídas digitais do arduino e da carta DS1104.

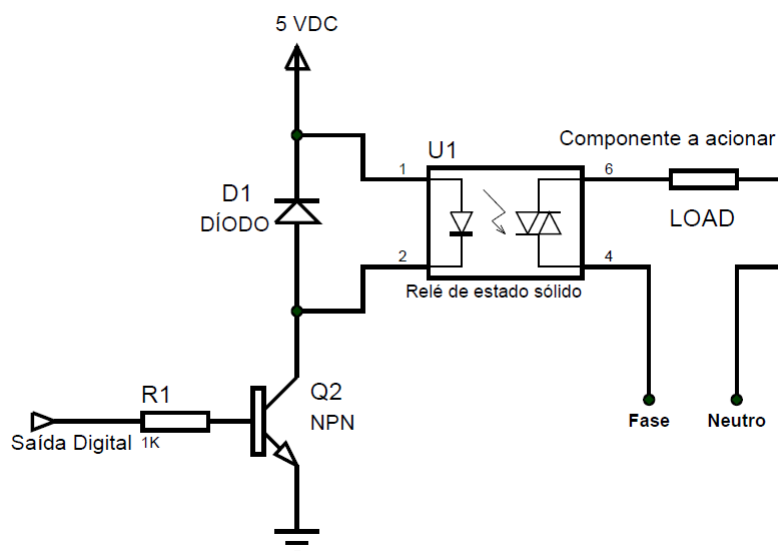


Figura 4.6: Circuito de acionamento dos atuadores de potência para a resistência e para a bomba de água

4.5.2 Condicionamento do sinal para o sensor de temperatura

O sensor LM35 é um sensor de temperatura que apresenta à saída uma tensão na ordem dos 10 mV/°C, com alimentação entre os 5 e os 30 VDC. Este apresenta-se vantajoso em relação a outros sensores de temperatura pelo facto de não necessitar de qualquer calibração externa. Como o sinal de saída do sensor é na ordem dos mV, optou-se por usar um amplificador de tensão para uma leitura mais ampla nas entradas analógicas do arduino e da carta de aquisição.

Usando um amplificador proporcional de ganho 10 conseguia-se uma melhor leitura no que diz respeito à escala, no entanto, um dos problemas deste tipo de medições é o ruído inerente a fenómenos externos. Existe uma grande flutuação do sinal que complica a componente de controlo e aquisição de dados. Pensou-se em usar como alternativa um amplificador diferencial. Se o sinal positivo for a leitura do sensor e o sinal negativo a sua massa, a diferença destes sinais elimina o ruído existente e permite uma leitura mais estável da temperatura. Pode-se observar na Figura 4.7 o esquema do amplificador aplicado.

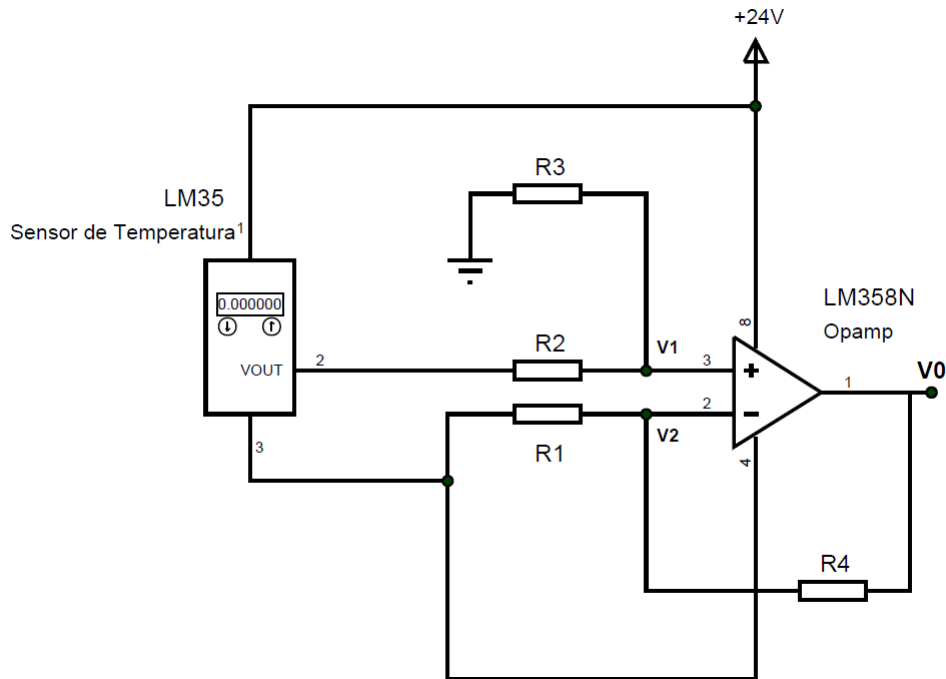


Figura 4.7: Esquema de amplificador diferencial do sinal de temperatura do LM35

Para saber qual o valor das resistências podemos observar a Equação 4.21.

$$Ganho = \frac{V0}{V1 - V2} = \frac{R4}{R1} = \frac{R3}{R2} \quad (4.21)$$

V1 e V2 são os valores, positivo e negativo que vêm do sensor, e V0 é o valor da tensão após a amplificação do sinal. Sendo o ganho 10 escolheu-se $R4 = R3 = 10000 \, \Omega$ e $R1 = R2 = 1000 \, \Omega$.

4.5.3 Condicionamento do sinal para o sensor de nível

No que concerne à detecção do nível do reservatório usou-se um sensor PNP de proximidade do tipo capacitivo, o RC4-3010F da RHOMBERG. Pretende-se que quando o sensor detete a presença de água a bomba seja desativada, e o oposto para quando deixa de detetar essa presença. Este pode ser alimentado de 10 a 60 VDC e 400 mA de corrente máxima. Este dispositivo, na presença de um corpo, a uma distância máxima de 10 milímetros, fecha os contactos permitindo a passagem de corrente de alimentação para a saída. Para adquirir esse sinal nas entradas digitais do *hardware* de controlo usou-se um simples divisor de tensão (Figura 4.8), visto que o sensor é alimentado a 24 VDC e se pretende que o sinal se aproxime dos 5 VDC.

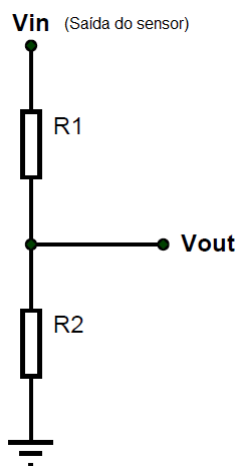


Figura 4.8: Esquema do divisor da tensão aplicada à saída do sensor de nível

Sabendo que se pretende um V_{out} de 5 VDC e que o V_{in} é de 24 VDC, através da seguinte equação determinam-se as resistências $R1$ e $R2$.

$$V_{out} = \frac{R2}{R1 + R2} \times V_{in} \quad (4.22)$$

Assumindo para a resistência $R1$ o valor de $2000 \, \Omega$ obtém-se, para $R2$, cerca de $500 \, \Omega$.

Assim chegou-se a toda a interface necessária para o condicionamento dos sinais que estarão no circuito da bancada que ligam o microcontrolador, a carta de aquisição de dados e os componentes do sistema de aquecimento. A Figura 4.9 representa um esquema simplificado do circuito elétrico, com os contactos para ligar os sinais entre os sistemas e uma representação dos módulos de condicionamento que foram mencionados.

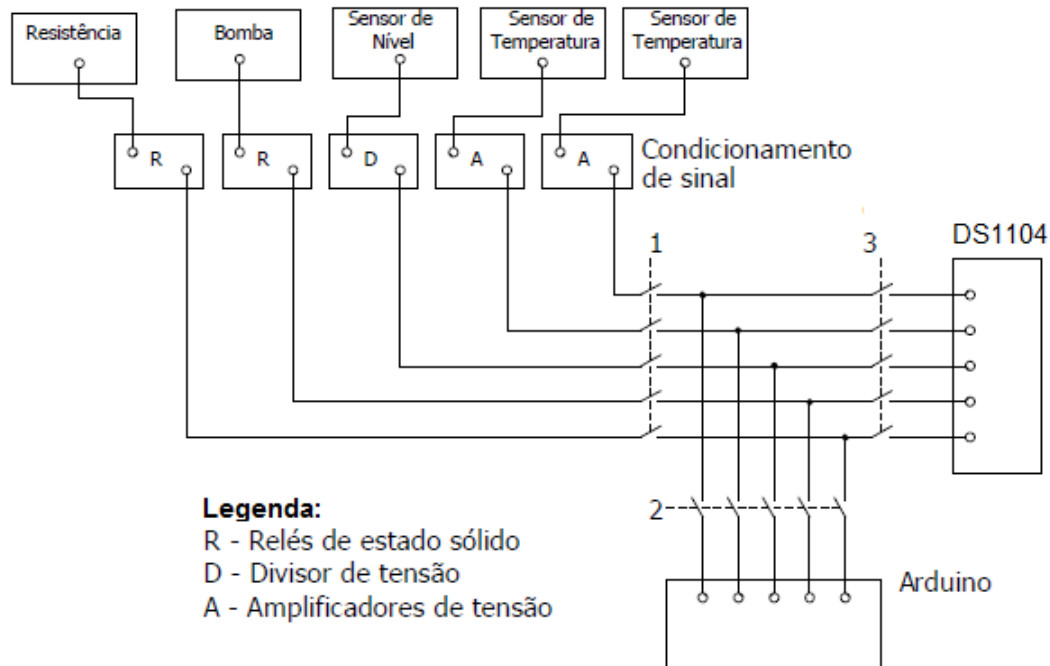


Figura 4.9: Esquema elétrico geral de funcionamento do caso de estudo

4.6 Adaptações no Simulink

De forma a que seja possível implementar esta ideia de bancada de ensaios flexível a vários tipos de testes, o próprio modelo Simulink precisa de ser adaptado para correr simulações em tempo real, comutar entre tipo de simulações e receber e enviar sinais do exterior e para o exterior. De seguida mostram-se algumas das principais alterações a ter em conta.

4.6.1 Adaptação dos sinais analógicos

Como já se referiu anteriormente os sinais analógicos a enviar e a receber correspondem à temperatura de entrada e do reservatório. Com o sensor LM35 a variação de temperatura corresponde a $10 \text{ mV}/^{\circ}\text{C}$ que, após o amplificador operacional, os sinais chegam na ordem dos $100 \text{ mV}/^{\circ}\text{C}$. Sabe-se que a carta de aquisição de dados, DS1104, envia e recebe sinais analógicos de 0 a 10 V.

A dSPACE associada ao *MATLAB/Simulink* tem uma biblioteca própria, a RTI Lib, que tem blocos para envio e receção de sinais. Neste caso esses blocos são os ADC ou DAC, para recepção e envio de dados, respetivamente.

Para envio dos sinais analógicos da temperatura simulada, úteis na situação em que o sistema virtual é controlado pelo arduino, utilizaram-se os blocos DAC Converter. Para isso, à temperatura simulada, efetuou-se a conversão necessária para se obter uma variação correspondente a 100 unidades por $^{\circ}\text{C}$. Após essa conversão efetuou-se a multiplicação por 10, para simular a amplificação do opamp. Assim, obteve-se o valor em Volt equivalente da temperatura, por exemplo para 30°C o valor a enviar tem de ser 3

V. No entanto, esta função necessita de uma adaptação final de forma a se obter o valor de tensão correto. Então o valor de tensão pretendido é necessário multiplicar na saída analógica da carta de aquisição de dados por 0.1 para a saída enviar o valor correto. A Figura 4.10 mostra os blocos utilizados, desde o valor obtido da temperatura simulada com a adaptação até enviar pelo bloco DAC para a carta de aquisição de dados. Ainda se observa um bloco de saturação para impedir que o valor ultrapasse os 5 V, devido às limitações de 0-5 V das entradas analógicas do arduino. No caso da temperatura de entrada da água o processo é semelhante.

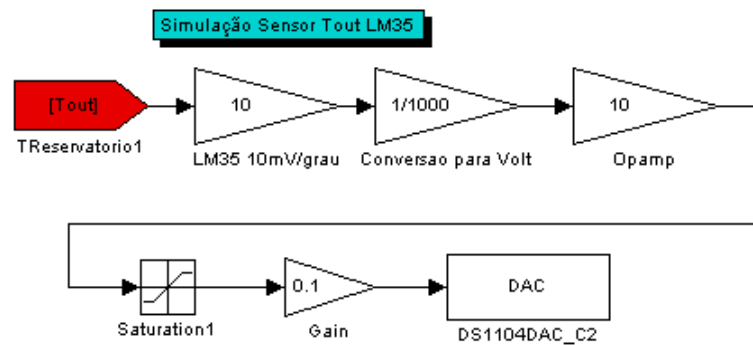


Figura 4.10: Conjunto de blocos para adaptação e envio do sinal de temperatura do reservatório pela saída analógica 2.

Para a recepção do valor de temperatura nas entradas analógicas da carta de aquisição de dados, necessário quando se está a controlar o sistema real pelo Simulink, é precisamente o inverso. É necessária a mesma adaptação e, para se obter o valor real de tensão, multiplicar a medição à saída do bloco ADC Converter por 10. A Figura 4.11 representa a implementação desses blocos para o caso da temperatura real do reservatório.

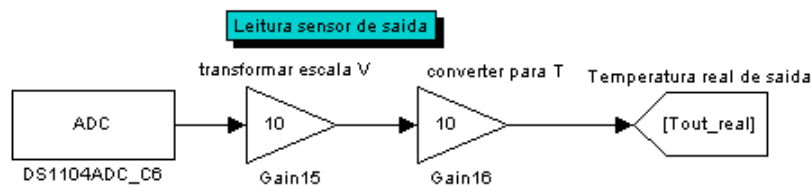


Figura 4.11: Conjunto de blocos para recepção e adaptação do sinal de temperatura do reservatório pela entrada analógica 6.

4.6.2 Adaptação dos sinais digitais

No que diz respeito a envio e recepção de sinais digitais, estes só interessam para o sensor de nível, o PWM da resistência de aquecimento e o sinal da bomba de água. É necessário enviar esses sinais digitais quando se está a utilizar o controlador virtual para controlar o sistema real, e recebê-los quando se pretende que o arduino controle o sistema virtual. A biblioteca do RTI 1104 disponibiliza no Simulink os blocos *Master Bit Out* e *Master Bit In*, para recepção e envio de sinais digitais, respetivamente.

No que concerne à receção dos sinais utiliza-se o Bloco *Master Bit In*, que toma o valor lógico 0 ou 1 caso receba 0 ou 5 V na entrada, respetivamente. Através desse valor lógico pode-se implementar no sistema as alterações pretendidas, seja para ligar a resistência, ativar bomba, ou outros.

Já para o correspondente envio de sinais digitais usa-se o bloco *Master Bit Out*. Para este também se tem que enviar valores lógicos de 0 e 1, de forma à saída digital ativar ou desativar o que se pretende.

A Figura 4.12 mostra dois exemplos simples de sinais, neste caso na receção e no envio do sinal de potência.

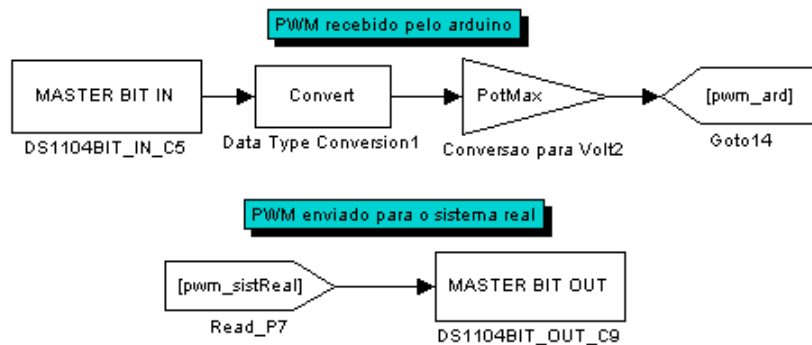


Figura 4.12: Conjunto de blocos para receção e envio do sinal PWM de potência da resistência

4.6.3 Comutação através do modelo virtual

Apesar de ser através dos relés de contacto auxiliar que é possível ligar os diferentes tipos de *hardware* entre eles, é também necessário adaptar o modelo em Simulink de forma a que consiga simular com dados internos ou externos. Esta preocupação justifica-se pelo facto de se pretender uma alteração mais rápida entre os tipos de simulação, e não haver a necessidade de conectar ou desconectar blocos do modelo.

Para implementar a distinção entre os diferentes tipos de simulação usou-se um *Multiport Switch*. Observe-se o exemplo da temperatura de entrada representado pela Figura 4.13.

Conforme se pode observar nessa figura, o utilizador apenas precisa de alterar a constante 'Tipo de simulação', com os valores 1, 2 ou 3, e a partir daí a variável "Tin" assume os diferentes parâmetros. Neste exemplo específico, caso se pretenda apenas simulação ou controlo do modelo pelo arduino a temperatura assume o valor 21; já se o objetivo for controlar o sistema real através do Simulink, a temperatura assume o valor da variável Tin real, que é um sinal exterior da temperatura de entrada no reservatório recebido pela carta de aquisição de dados.

Não só para a temperatura de entrada é necessária esta alteração. Para todos os restantes sinais que são enviados ou recebidos recorreu-se a um *Multiport Switch*. É o caso da temperatura do reservatório, do sinal do sensor de nível, do sinal da bomba e do sinal PWM de potência. Para todos existe uma variável principal, que assume os diferentes valores dependendo da simulação em questão. É essa variável que será utilizada nos cálculos do modelo e do controlador.

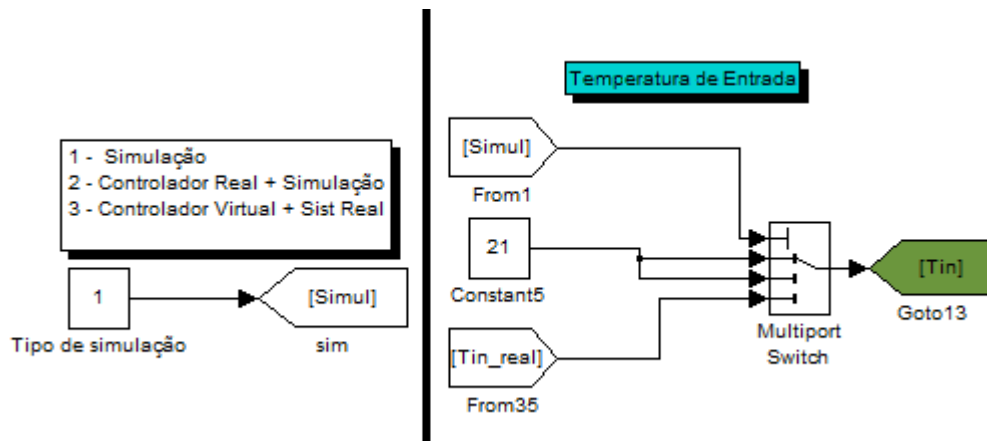


Figura 4.13: Esquema de comutação entre simulações no Simulink

Com as alterações referidas é possível tornar a transição entre simulações mais rápida, sem serem necessárias mudanças de ligações a cada opção de simulação. Se o modelo for bem estruturado consegue-se adaptar rapidamente qualquer tipo de sistema, usando os blocos adequados, para que receba e envie sinais para o exterior.

4.7 Interface e monitorização no ControlDesk

Como já se referiu anteriormente, para monitorizar e controlar a simulação em tempo real é necessário recorrer a um *software* próprio que, no caso da dSPACE, é o *ControlDesk*.

Com o *ControlDesk* é possível alterar estados da simulação, efetuar a monitorização dos sinais e inclusive adquirir os dados. Decidiu-se colocar na interface todas as variáveis que dizem respeito a temperaturas, caudal, estado do nível e todos os níveis de tensão recebidos nas entradas e saídas da carta da dSPACE.

No que diz respeito a parâmetros de controlo que podem ser modificados aquando da simulação, é possível alterar o tipo de simulação e a temperatura desejada, estando as restantes variáveis bloqueadas.

A nível de aquisição de sinais recorreu-se ao *Data Acquisition*, disponível no *ControlDesk*, que permite gravar para um ficheiro '.mat' as variáveis que se pretendam, que no caso foi a temperatura, o sinal PWM e o estado da bomba. Através desta opção é possível documentar as alterações principais no sistema para posterior tratamento de dados e construção de gráficos.

São inúmeras as vantagens deste *software*, pois permite a partir de um só programa monitorizar, de forma intuitiva e gráfica, todo o processo de testes, simulado ou não.

Encontra-se no Apêndice C o *layout* utilizado no *ControlDesk* deste caso de estudo.

4.8 Programação do arduino

Outra das componentes importantes deste trabalho é a programação do controlador real. Este é responsável pelo controlo do sistema real e pelo controlo do sistema virtual, quando

solicitado para isso. O material disponibilizado para esta componente foi um *arduino uno* que, para o caso de estudo pretendido, é suficiente.

O *arduino* tem como entradas digitais o sensor de nível e como saídas digitais o atuador para a bomba de água e para a resistência de aquecimento. No que concerne a entradas analógicas tem os sensores de temperatura, que podem ser do sistema real ou os simulados no sistema virtual. A Figura 4.14 mostra as ligações realizadas no *arduino uno* com as entradas e saídas digitais ("DI " e "DO ") e as entradas analógicas ("AI ").

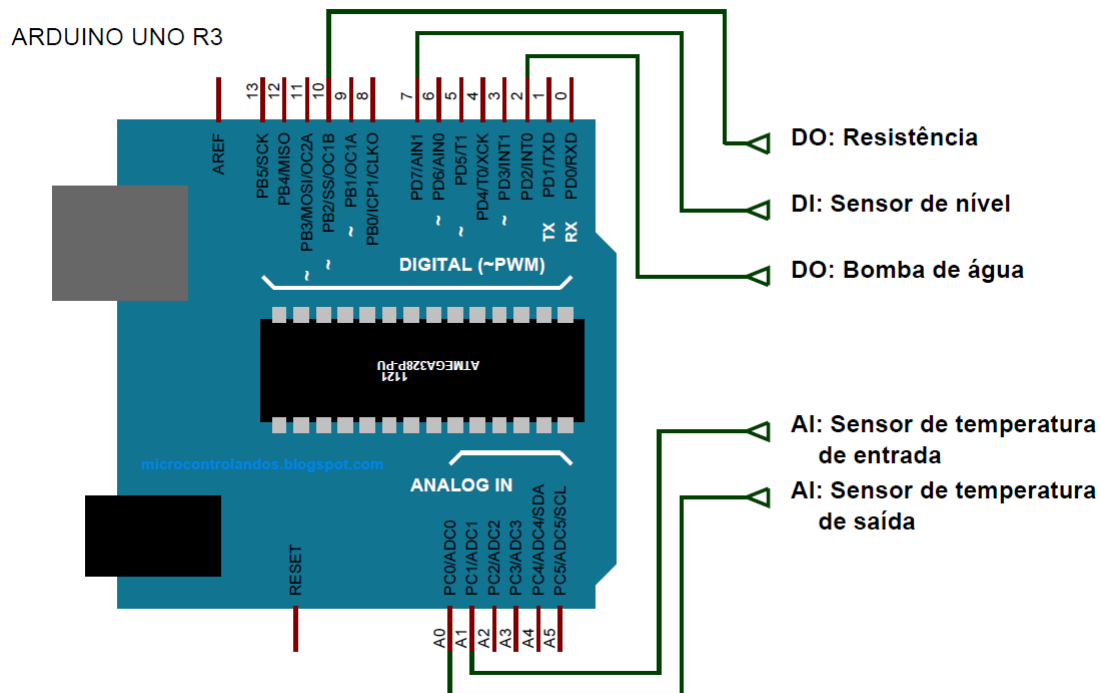


Figura 4.14: Representação do *arduino uno* com as ligações utilizadas no caso de estudo

O princípio de funcionamento do controlo implementado no *arduino* é semelhante ao do controlador virtual do Simulink referida na Secção 4.3. Este calcula continuamente a potência necessária a dar ao sistema, com base nas temperaturas de entrada e saída, e dá uma resposta através do controlador PID. Quando deteta que a entrada digital do sensor de nível foi ativada, desliga a saída digital da bomba e volta a ligá-la assim que o sensor desligar, para que mantenha a coluna da água constante. Importa salientar que o próprio sensor de nível tem histerese, o que provoca o tempo de espera para que o sensor passe do estado *on* a *off* e vice-versa.

A estrutura do programa no *arduino* é dividida pela parte inicial, onde se declaram as variáveis do programa, pelo *setup*, que é onde se fazem algumas das configurações necessárias e pelo *loop*, local onde fica o código principal do programa que está continuamente a correr em ciclo e contém todas as partes de aquisição de sinal e cálculo de potência. Para leitura da entrada analógica com o sinal do sensor de temperatura foi implementado, no código principal, um ciclo que a cada chamada lê várias vezes a temperatura e no fim calcula a sua média. Esta opção serviu para limitar a hipótese do uso de valores de temperatura pouco reais na parte do controlo, provocadas por eventual ruído, o que influenciaria o cálculo da potência.

No que diz respeito ao controlo de temperatura foi implementada a equação de potência e o respetivo controlador, tal como no sistema virtual. A cada ciclo é calculado o valor de potência teórica e efetiva com a temperatura que está a ser lida nas entradas analógicas. De seguida determina-se o *Duty Cycle*, do sinal PWM a ser gerado, dividindo a potência efetiva pela potência máxima da resistência, que dá basicamente a percentagem que se pretende da potência máxima. A nível do cálculo do PID recorreu-se à biblioteca "PID_v1.h". Esta aceita como argumentos o valor de cada um dos ganhos, proporcional, derivativo e integral, e os valores de entrada, referentes à potência efetiva e teórica. Como saída tem o valor da potência de resposta já com o cálculo do PID. É necessário ajustar alguns parâmetros como o *sample rate* e gravar em variáveis o valor anterior de tempo e temperatura.

No que concerne ao sinal PWM de potência algumas das possibilidades passavam pela utilização dos *timer interrupts* e a geração direta através das saídas PWM do *arduino*; no entanto, têm desvantagens associadas, como é o caso da dificuldade em manipular a frequência do sinal. Esta é possível de ser alterada mas para valores tabelados diferentes da frequência pretendida, de cerca de 20 Hz. De forma a contornar essa situação utilizou-se a biblioteca "TimerOne.h", que usa o Timer 1 do *arduino*. A vantagem dessa biblioteca é o facto de permitir modelar a frequência de tempo do Timer1. Os Timers são relógios internos do *arduino* que podem ser usados para correr conjuntos de código de forma sincronizada e repetitiva. Assim, cada vez que o relógio executa uma interrupção, chama uma função que altera o estado da saída digital para 1 e 0, gerando o sinal PWM. Para ser possível manipular as percentagens de tempo a 1 e a 0 alterou-se a cada ciclo o valor da frequência de sinal, de forma a que a soma total do tempo a 1 e a 0 corresponda a uma frequência de 20 Hz.

O princípio de funcionamento do controlador é o mesmo que o projetado virtualmente no *MATLAB/Simulink*. É recebido, conforme foi referido anteriormente, o valor de temperaturas que é usado para o cálculo da potência necessária ao sistema e para o cálculo do controlador para a potência necessária a ser fornecida.

O *arduino*, quando deteta o sinal do sensor ligado, desliga a saída da bomba de água para controlo do nível e vice-versa, para manter o nível da água constante.

O Apêndice B mostra o código que foi implementado no *arduino* para o controlo do sistema em causa.

4.9 Montagem experimental

Com base nas pesquisas efetuadas referente a *hardware-in-the-loop* e a bancadas de ensaios, após a definição do caso de estudo e da elaboração do projeto elétrico foi elaborada uma lista cuidada do material necessário. Houve este cuidado visto que algum material teve que ser adquirido para este fim específico, e para avançar de forma organizada e estruturada. No que diz respeito ao *software* e *hardware* de simulação em tempo real, foi disponibilizado uma carta DS1104 R&D Controller Board e o Release 4.0 da dSPACE com todo o *software* de simulação. No que concerne ao *software* de modelação recorreu-se, como já foi abordado, ao *MATLAB/Simulink* 7.0.4, imposto pelos requisitos da dSPACE. O computador para modelação e monitorização do processo foi instalado com o Windows XP e com um *upgrade* de memória para 4 GB de RAM.

A nível de controlador real recorreu-se a um *arduino uno*. Usou-se esta opção pelo

facto de não ser necessário comprá-lo, e tendo em conta que o seu funcionamento se assemelha a um microcontrolador, que serve perfeitamente para os testes pretendidos. Para o reservatório de água, após o esquematizar inicialmente, recorreu-se primeiramente ao modelo em *MATLAB/Simulink* de forma a simular várias áreas de reservatório, com determinada coluna de água e caudal de saída. O objetivo foi verificar se o caudal não ultrapassaria o da bomba de entrada, e saber se a resistência teria potência suficiente para aquecer a quantidade de água presente. Concluiu-se então que um reservatório em acrílico com dimensões de $250 \times 250 \times 250 \text{ mm}^3$ [15.6 L] e espessura de 6 mm, seria suficiente para realizar as experiências pretendidas.

A Tabela 4.6 resume os restantes componentes mais associados à parte de condicionamento de sinal e sensorização, tendo alguns sido adquiridos especificamente para este fim. Importa referir que os sensores LM35 foram isolados de forma a poder ser mergulhados em água, visto que não são apropriados para o efeito. O projeto elétrico efetuado previamente foi importante na medida em que possibilitou o planeamento e compra apenas do material necessário

Tabela 4.6: Lista de material associado aos componentes elétricos, de atuação e sensorização

Material utilizado	Qtd.
Relé de contacto Omron G2RG2A412DC	12
Relé de estado sólido G3MC-202PL DC5	1
Relé de estado sólido Shimaden PAC03B	1
Resistências elétricas	12
Transistor PNP	2
Amplificador LM358N	2
Díodo	2
Sensor de temperatura LM35N	2
Fonte de alimentação	1
Resistência de aquecimento 2000W	1
Bomba de água	1
Sensor capacitivo	1

Com todo o material definido e disponível efetuou-se a montagem experimental com sensores, atuadores, ligações entre sistemas, montagem dos módulos de interface e alimentação de todo o circuito. Pode-se observar na Figura 4.15 toda a montagem experimental do sistema em teste. A Figura 4.16 mostra em pormenor o sistema real com o reservatório, e sistema de atuação e sensorização instalado. Por último, a Figura 4.17, representa mais em detalhe o controlador real, o sistema virtual e todos os módulos de condicionamento projetados, bem como os circuitos e componentes utilizados.

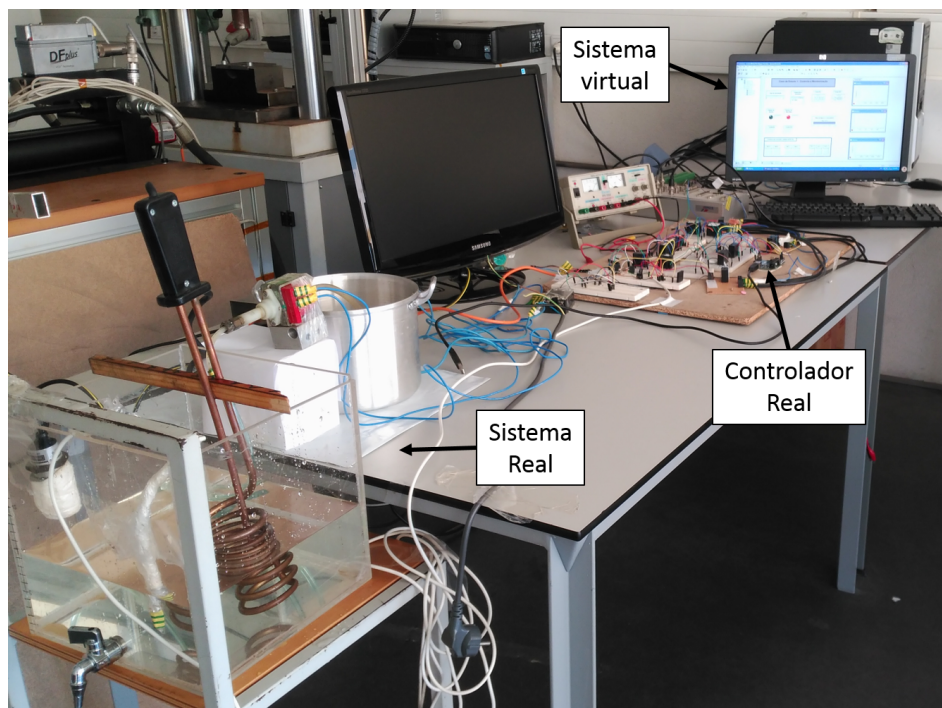


Figura 4.15: Montagem de todo o sistema real, virtual e controlador real para os testes

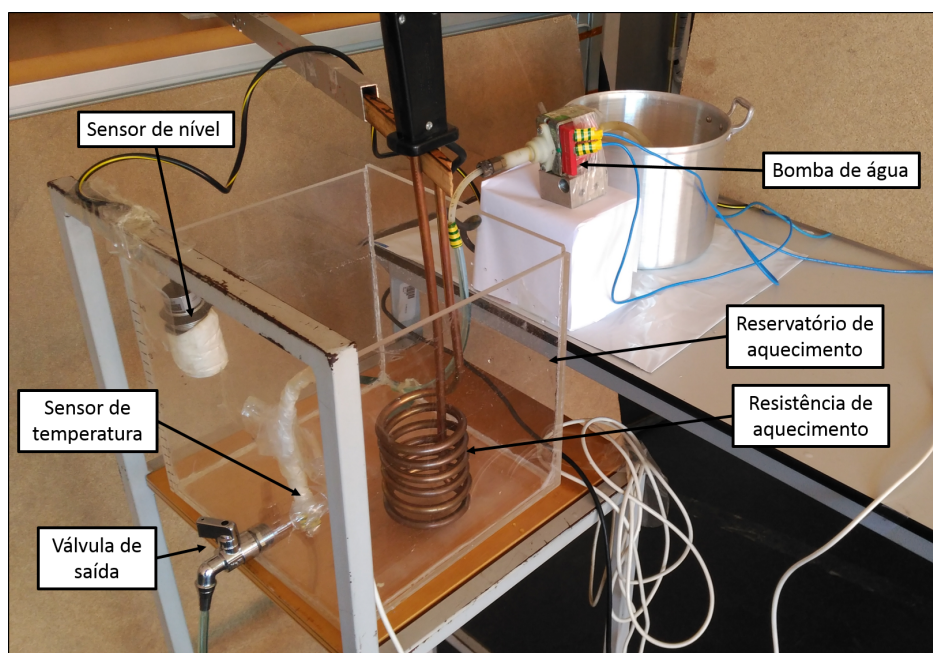


Figura 4.16: Montagem do sistema real correspondente ao reservatório e respectivos atuadores e sensores

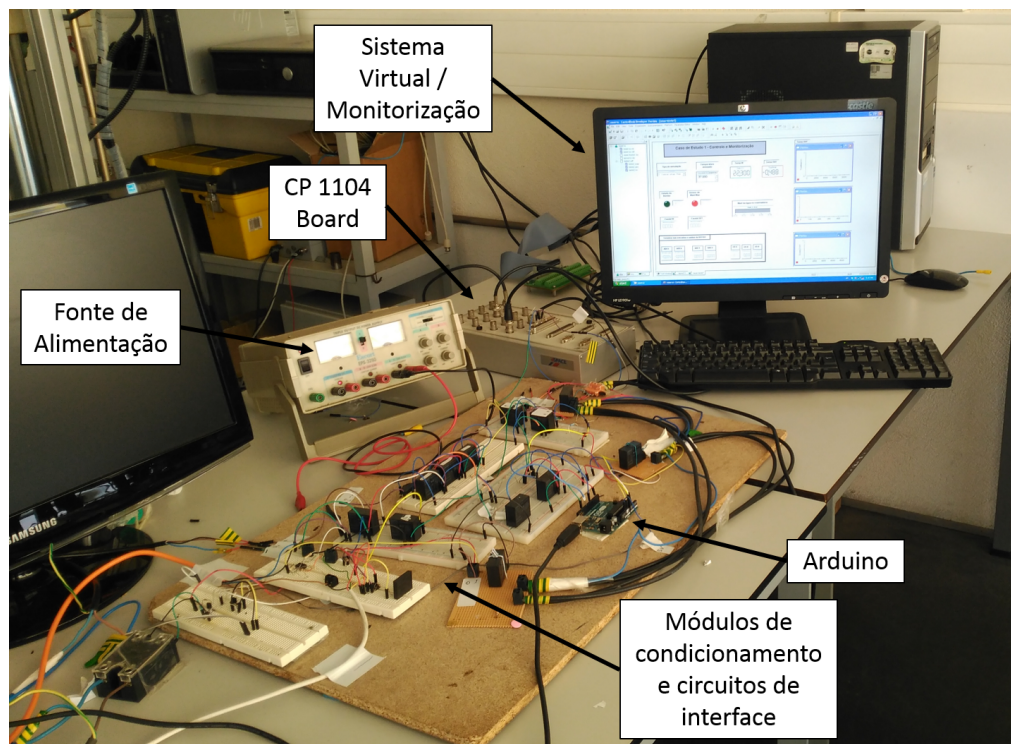


Figura 4.17: Montagem com os circuitos de condicionamento, *arduino* e *hardware* do sistema virtual

4.10 Procedimento experimental geral

Após realizada toda a montagem experimental procedeu-se à parte correspondente aos testes. Para realizar essa componente prática há um conjunto de procedimentos que tiveram que ser seguidos para o bom funcionamento do sistema. Nesta secção apenas se indica a ordem seguida para efetuar um determinado teste, tendo em conta a forma como o modelo foi desenvolvido e como se comutam os relés para cada ensaio.

No que concerne aos testes do controlador virtual no modelo virtual, praticamente não há alterações. É necessário deixar o botão de comutação na opção D, de forma a abrir todos os contactos dos relés. O tipo de simulação tem de estar na opção 1, que automaticamente passa a simular o sistema sem qualquer sinal externo. Nesta parte a aquisição de dados pode ser efetuada através do *ControlDesk*, compilando o modelo *Simulink* para a dSPACE 1104, ou então diretamente do *MATLAB/Simulink* gravando os dados da temperatura num ficheiro, que nesta experiência específica foi o mais utilizado.

Para ensaiar o controlador virtual no sistema real é necessário antes de mais alterar os sinais, de forma a que o controlador receba a temperatura nas entradas analógicas, receba o sinal do sensor de nível nas entradas digitais e envie o PWM e o sinal da bomba pelas saídas digitais. É preciso compilar o modelo e monitorizar todos os passos pelo *ControlDesk*. De seguida coloca-se o tipo de simulação na opção 3 e o botão de comutação na posição C. A aquisição de dados é feita no "Data Acquisition" para um ficheiro ".mat" a cada 5 s.

No que diz respeito ao controlo do sistema virtual pelo *arduino*, inverteram-se os sinais do anterior, enviando a temperatura e o sensor de nível e recebendo a potência e o sinal de bomba. Além disso, colocou-se na opção de simulação 2 e o botão de comutação na posição B. Os passos para aquisição de dados são iguais ao anterior. Por último, para a possibilidade de controlo do sistema real pelo *arduino*, alterou-se o botão de comutação para a posição A. Deixou-se fechado o contacto correspondente à temperatura na carta de aquisição de dados para a monitorizar no *ControlDesk*, da mesma forma que os anteriores.

Para todos os processos ajustaram-se os valores de caudal, temperatura ambiente e temperatura de entrada da água. Houve a preocupação de, em primeiro lugar, medir experimentalmente cada uma dessas variáveis para ajustar o modelo virtual e os controladores de forma a que os ensaios fossem feitos nas mesmas condições.

Para uma maior lógica cronológica dos testes, o objetivo passou por manter os passos típicos no processo de desenvolvimento do produto e tentar aproximar a ordem dos testes ao que poderia acontecer no mundo industrial. Assim, primeiramente tentou-se validar o modelo através de testes de comparação ao sistema real. De seguida, através de testes ao sistema virtual e real, projetar o controlador de forma a ficar mais eficiente. Com o controlador aprovado, implementá-lo no *arduino* e testar no sistema virtual, de forma a verificar se os controladores, virtual e real, se aproximam nos testes ao modelo. Por último, testar o comportamento do sistema real através do *arduino* e verificar se responde conforme o desejado.

De seguida descrevem-se com maior pormenor cada uma das experiências realizadas, com os respetivos resultados e conclusões.

4.11 Parametrização do modelo virtual

Os sistemas virtuais pretendem traduzir num simulador aquilo que se passa na realidade. Obviamente que estes sistemas apenas se comportam consoante as variantes que forem impostas por quem o desenvolve. No entanto, há sempre variáveis complicadas de modelar e determinar que tornam o sistema virtual diferente do real. Além disso, o sistema desenvolvido em *Simulink* considera que a temperatura é igual em todos os pontos do reservatório, o que na realidade não acontece. Outro dos pormenores relevantes neste modelo em questão é o transporte de massa no interior do reservatório, que não foi considerado. Porém, como a perfeita modelação do sistema não é um dos objetivos desta dissertação, apenas se vai verificar o comportamento de ambos os sistemas e se há a necessidade de ajustar alguns parâmetros.

Recorreu-se a uma parametrização do modelo através de testes experimentais para determinar quais as principais diferenças entre ambos os sistemas. Esta é outra das grandes vantagens apresentadas no conceito de bancada apresentado, o facto de com o mínimo de alterações possível se comutar entre o sistema virtual e o real.

O objetivo passou por aplicar gamas de potência em primeiro ao sistema real, com determinadas condições de temperatura ambiente, e de seguida inseri-las no modelo virtual e simular da mesma forma. A aquisição dos dados para posterior análise ficou a cargo do *ControlDesk*.

Antes disso observe-se a Tabela 4.7, que resume todas as variáveis físicas constantes, independentemente dos testes, definidas nos cálculos de cada um dos elementos do *Simulink*.

Tabela 4.7: Resumo das propriedades constantes usadas no sistema virtual e controlador

Parâmetros do sistema	Valor	Unidades
Calor específico da água	4186	J/kg.K
Massa volúmica da água	1000	kg/m ³
Condutividade térmica do acrílico	0,19	W/m.°C
Coefficiente de convecção para o ext.	15	W/m ² .K
Caudal de entrada de água	0.86	L/min

4.11.1 Testes de comportamento

Para determinar o comportamento do sistema real definiram-se em primeiro lugar três gamas de potência constante de 50, 75 e 100 % da potência máxima da resistência. Pretendeu-se não ter caudal de entrada nem saída e apenas verificar a variação da temperatura pelas perdas de calor e pela potência térmica aplicada. Efetuou-se esse teste durante 300, segundos com a temperatura da água inicialmente igual à temperatura ambiente.

Para monitorizar o sistema real mudou-se o botão de comutação da bancada de ensaios ligando o sistema real ao sistema virtual, posição C. Assim, monitorizou-se a temperatura através do *ControlDesk* e gravou-se os dados no computador durante 300 segundos. A carta de aquisição de dados recebeu nas entradas analógicas a temperatura e enviou pelas saídas digitais o sinal de PWM, gerado através do *Simulink*.

De seguida, abrindo os contactos entre os sistemas, passou-se para a parte de simulação. Assim, na vez do *Simulink* aplicar essa potência para o exterior foi aplicada no próprio modelo virtual, sem o envolvimento de outro tipo de *hardware* exterior.

Em primeiro lugar houve a preocupação de deixar os argumentos de temperatura ambiente e temperatura inicial da água semelhantes aos obtidos experimentalmente. Além disso, as condições de caudal foram as mesmas, ou seja, sem caudal de entrada nem de saída, mantendo a coluna de água constante. Depois, iniciaram-se os testes aplicando a mesma potência que foi aplicada no modelo real, durante 300 segundos, e registou-se o comportamento da temperatura

Verificou-se pelas Figuras 4.18, 4.19 e 4.20 os pares de resultados para cada gama de 50, 75 e 100 %, respetivamente, da potência máxima para o sistema real e virtual.

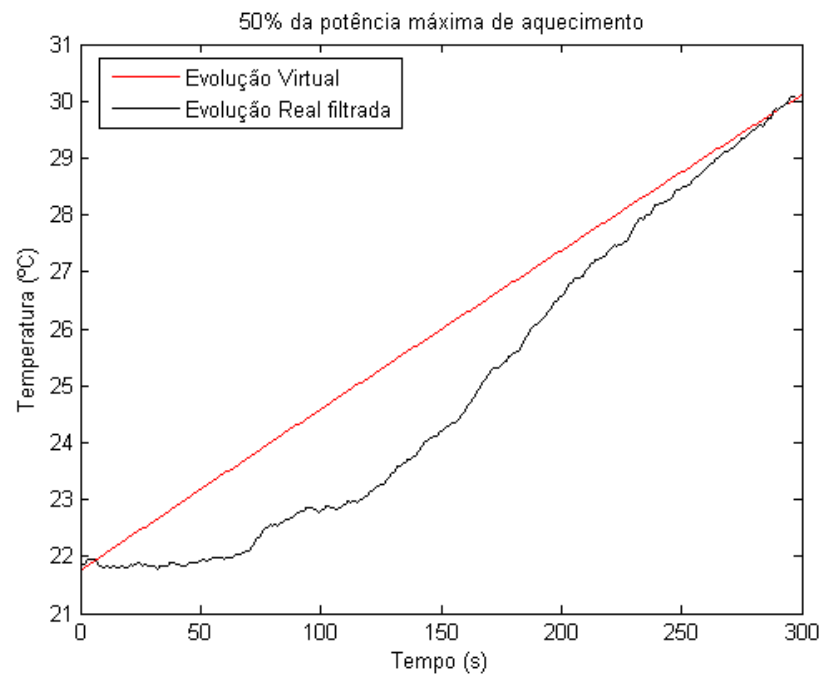


Figura 4.18: Evolução da temperatura com 50 % da potência máxima para o sistema real e virtual

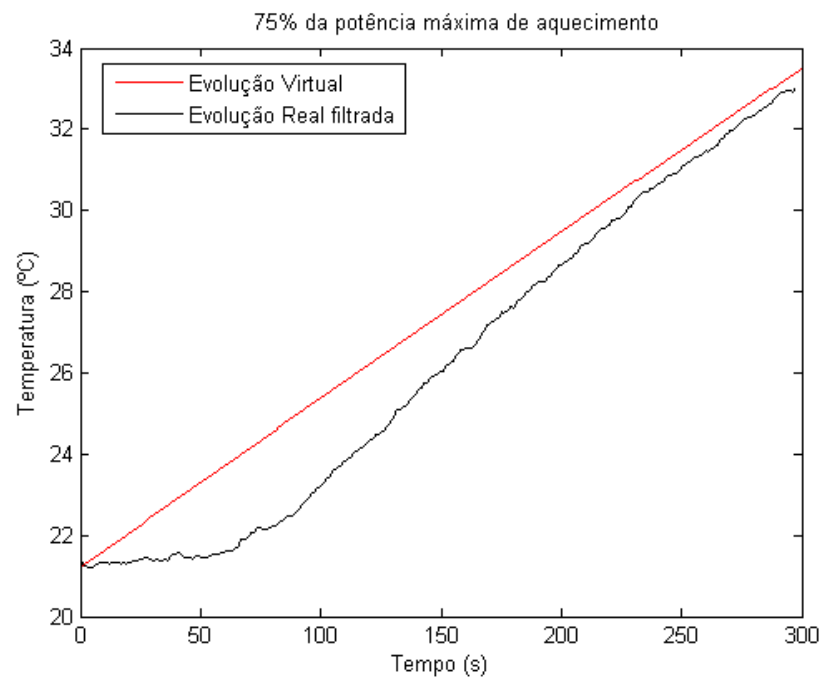


Figura 4.19: Evolução da temperatura com 75 % da potência máxima para o sistema real e virtual

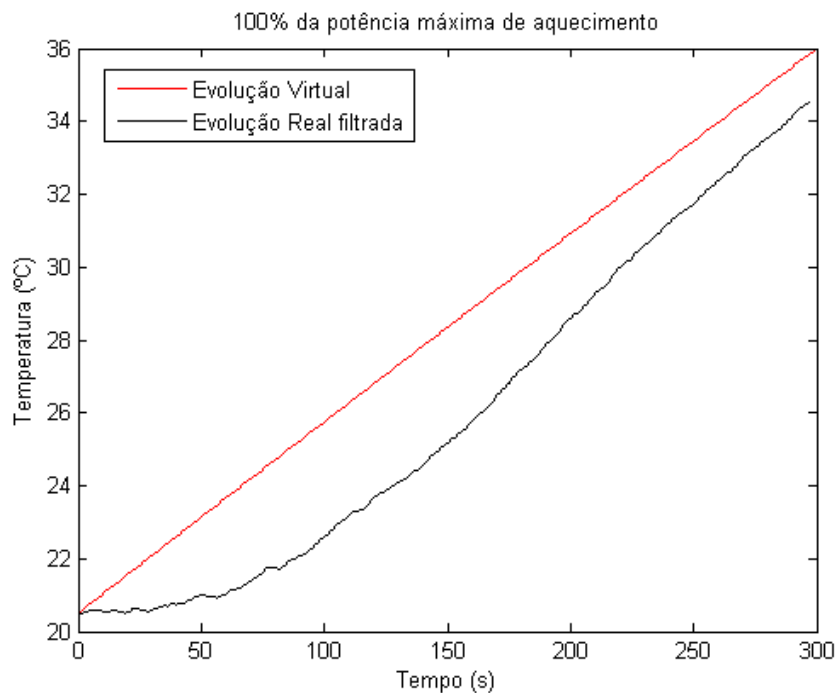


Figura 4.20: Evolução da temperatura com 100% da potência máxima para o sistema real e virtual

Observou-se que, com os parâmetros inseridos, a temperatura final de simulação quer do sistema real quer do sistema virtual tendiam para um valor próximo, com uma variação de temperatura semelhante após um determinado tempo. Observa-se que o sistema virtual responde logo com um aumento de temperatura, enquanto que o sistema real vai mantendo a temperatura constante até iniciar a sua subida. Esta diferença pode ser justificada pela simplificação do modelo simulado. Este assume que a temperatura é igual em todos os pontos do reservatório e que a potência é aplicada da mesma forma. O que não acontece no sistema real, pois, existe um tempo de espera inicial até o sensor de temperatura receber o seu aumento. Isto deve-se ao fluxo de calor, tendo em conta que a potência é aplicada numa zona muito restrita, e que a temperatura apenas vai aumentando aos poucos na vizinhança.

Seria necessário adaptar o sistema virtual de forma a que se aproxime dos valores que ocorrem na realidade. Ainda assim, o que importa nesta situação é comprovar o conceito de bancada de ensaios e não a perfeita modelação do sistema. Por essa razão passa a ser praticamente impossível de obter resultados suficientemente aproximados entre ambos numa fase inicial. No entanto, como o gráfico da variação da temperatura virtual tem um comportamento semelhante à variação da temperatura real ao fim de algum tempo pode-se considerar que o modelo serve para o tipo de testes em questão.

4.12 Projeto do controlador

O projeto do controlador, tal como já foi referido anteriormente, foi feito de forma aproximada. Durante simulações consecutivas testaram-se diferentes valores de parâmetros PID para o controlador, até encontrar os melhores valores.

Apesar deste método não ser o ideal para o projeto de controladores, é uma forma simples de os parametrizar. Como o sistema em questão não é muito complexo e o projeto ideal do controlador não é objetivo desta dissertação, considera-se a estratégia adequada e suficiente para o fim em vista.

Para isso, passou-se o sistema para o modo 1 de operação, fazendo com que apenas haja interação entre o controlador e o sistema em ambiente simulado.

No que diz respeito aos parâmetros do controlador usados para o sistema estabilizar nos 35 °C, estes foram determinados por tentativa erro no sistema virtual. Definiram-se as temperaturas de entrada de água e ambiente, medidas antes da experimentação, sendo a primeira de 22 °C e a segunda de 23.5 °C.

No que concerne a entrada e saída de água, optou-se por manter o caudal de saída constante e existir caudal de entrada quando o sensor de nível deixar de detetar a água. Para caudal de entrada tem-se cerca de 0,86 L/min, enquanto que o caudal de saída foi medido experimentalmente com a abertura da válvula constante, bem como a coluna de água, obtendo-se o valor de 0,34 L/min.

Chegou-se assim aos valores da Tabela 4.8 para cada uma das componentes proporcional, integral e derivativa.

Tabela 4.8: Valores de PID determinados por tentativa-erro.

Tipo de controlador	Kp	Ki	Kd
P	10	-	-
PI	9	216	-
PID	12	130	32

O comportamento da temperatura do sistema com cada um dos controlos referidos na tabela anterior estão representados na Figura 4.21.

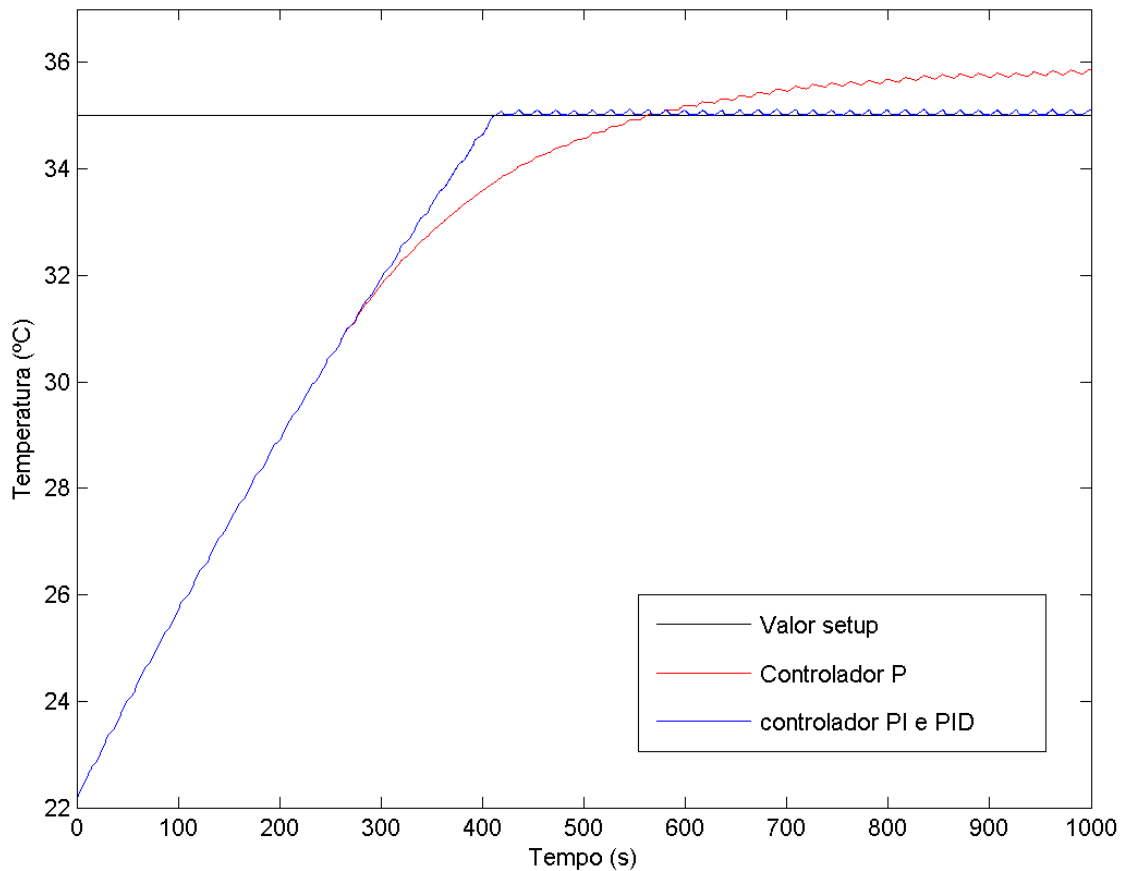


Figura 4.21: Teste a cada um dos controladores calculados P, PI e PID

Observa-se que inicialmente ambos os controladores têm o mesmo comportamento sobrepondo-se. O controlador P ao chegar perto da temperatura de *setup* vai aumentando a temperatura de forma mais linear e lenta, estabilizando aproximadamente um grau acima do desejado. Já o PI e o PID estão praticamente sobrepostos e aproximam-se de forma quase ideal da temperatura definida, estabilizando nesse ponto.

Tendo em conta os resultados a opção recai no controlador PI, sabendo que é suficiente para controlar o sistema, não havendo necessidade de aplicar uma componente derivativa.

4.13 Testes de controlo

Construídos todos os circuitos, montado o equipamento e modelado todo o sistema, chega a altura de comprovar que o conceito proposto funciona. Pretende-se efetuar testes de controlo de temperatura com as variantes do modelo e retirar os resultados de cada uma. O objetivo passa por encenar as etapas de projeto de um sistema térmico, efetuando os passos que se consideram comuns a qualquer um. Dessa forma, seguindo uma linhagem lógica, em primeiro efetuaram-se os testes apenas em ambiente simulado, ou seja, aquando o projeto do controlador, onde se verifica o comportamento do sistema modelado. De seguida, com o protótipo construído, testa-se o controlador projetado em *Simulink* no sistema real e verifica-se se a temperatura é controlada como desejado, efetu-

ando os ajustes necessários na parametrização do modelo. Após os ajustes e otimização do controlador programa-se o *arduino* e efetuam-se testes em ambiente simulado. Por último, após a validação dos testes anteriores, efetua-se um último ensaio do *arduino* a controlar o sistema real.

De seguida mostram-se os passos efetuados em cada teste, com os respetivos resultados e ajustes realizados.

4.13.1 Teste do controlador virtual no modelo virtual

Tal como se referiu anteriormente, após toda a modelação, parametrização e validação do modelo é necessário verificar se o controlador projetado tem o comportamento desejado. Dessa forma coloca-se o botão de seleção na opção D, desativando todos os contactos no circuito. Além disso, é preciso adaptar o *Simulink* para o tipo de simulação, que é desligando todos os sinais exteriores e este apenas interagir com os simulados.

Definiram-se os valores de caudal de entrada e de saída, semelhantes ao que aconteceria na realidade, a altura da coluna de água inicial e a altura máxima e mínima para o qual o sensor de nível ativa e desativa a entrada de água. Todos esses dados foram retirados do protótipo para maior aproximação à realidade. Foi necessário, além disso, parametrizar todas as variáveis exteriores comuns como é o caso da temperatura ambiente e da temperatura da água à entrada, valores estes medidos experimentalmente. A Tabela 4.9 resume os parâmetros inseridos no modelo:

Tabela 4.9: Parâmetros alterados no modelo virtual, com base nas medições experimentais, para a simulação

Parâmetro do sistema	Valor		Unidades
Temperatura ambiente	23,4		°C
Temperatura da água à entrada	22		°C
Caudal de saída	0,34		L/min
Sensor de nível	On	0,139	m
	Off	0,140	

Com todos os parâmetros inseridos, simulou-se o reservatório de água como se fosse um reservatório para fornecer água para o utilizador tomar banho. Assim, definiu-se a temperatura de *setup* cerca de 37°C. A Figura 4.22 mostra os resultados da temperatura obtida enquanto que a Figura 4.23 apresentam-se os valores para um foco maior na zona onde a temperatura estabiliza.

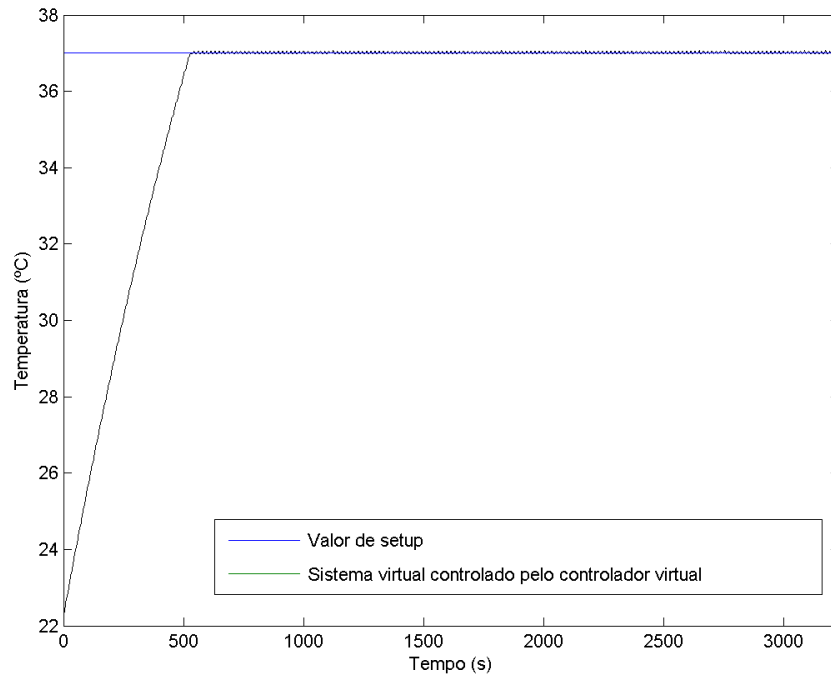


Figura 4.22: Teste do controlador virtual no sistema virtual com T_{setup} de 37°C.

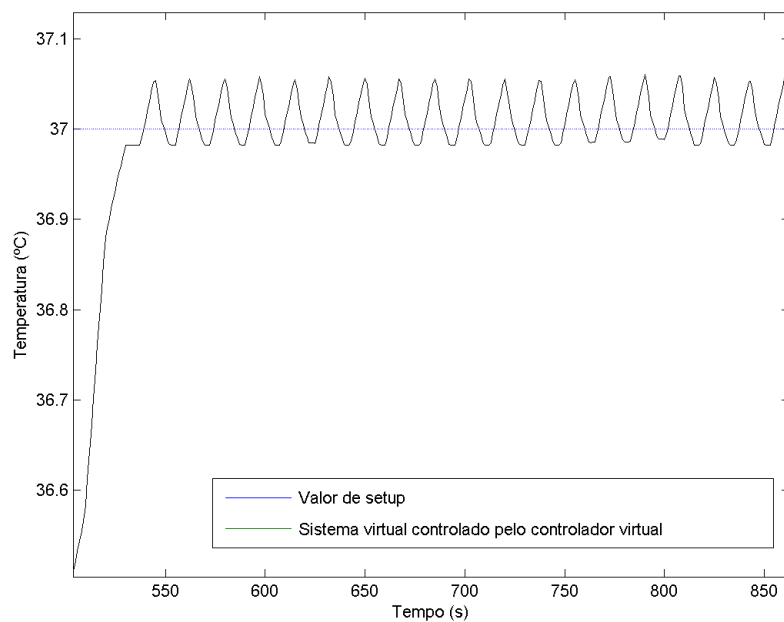


Figura 4.23: Teste do controlador virtual no sistema virtual com T_{setup} de 37°C, com maior foco na zona de estabilização.

4.13.2 Teste do controlador virtual no modelo real

Com os ensaios do controlador virtual no sistema modelado realizados e com a validação do controlador, seguiram-se os testes no sistema real. Este passo é útil na medida em que se pode testar o comportamento do controlador diretamente no protótipo, sem a necessidade de se programar o *arduino* em primeiro lugar. Apesar do controlador funcionar perfeitamente no sistema virtual, é natural que se comporte de maneira diferente quando controla sistemas reais e sem condições ideais de funcionamento. Neste ensaio os valores introduzidos no modelo não foram alterados, e os restantes dados foram medidos diretamente pelos sensores, como é o caso da temperatura e do sensor de nível.

Para a experiência altera-se o botão de comutação para a posição C, ligando a carta de aquisição de dados ao sistema real. Além disso, fazem-se as adaptações necessárias para que o controlador virtual desta vez receba os sinais do sensor de nível, e das temperaturas vindas do exterior, e envie o sinal para ligar a bomba e o PWM para a resistência. É possível verificar os resultados obtidos nas Figuras 4.24 e 4.25, onde a temperatura medida no reservatório foi filtrada no MATLAB para suavizar algum ruído.

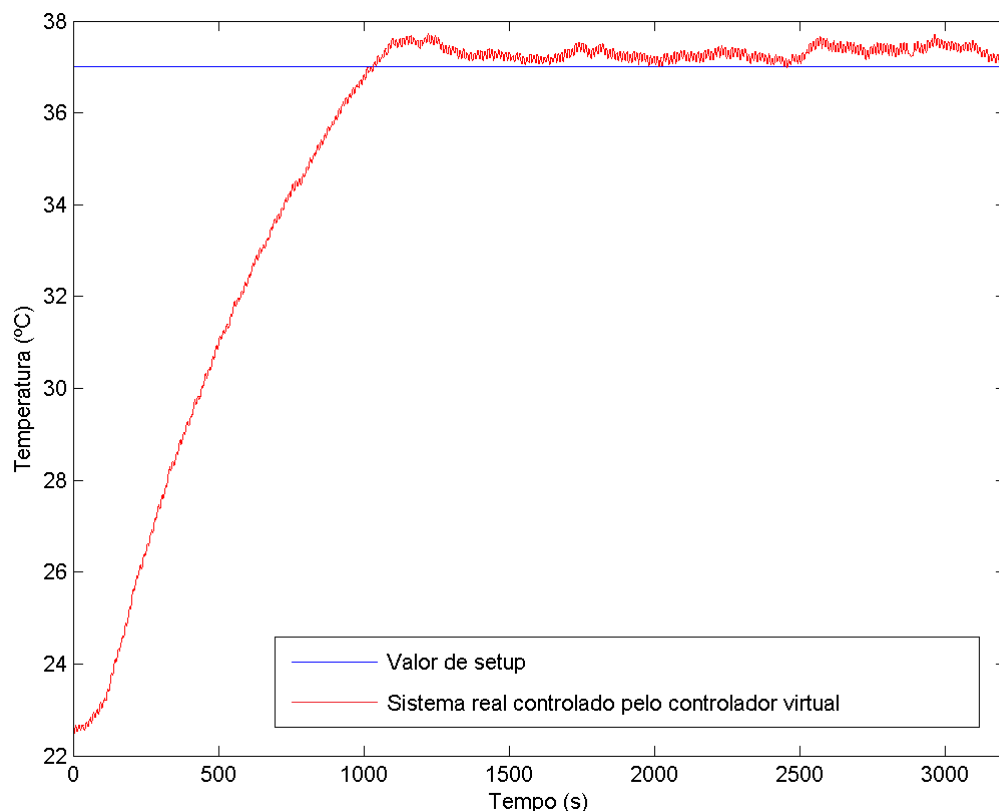


Figura 4.24: Teste do controlador virtual no sistema real com T_{setup} de 37°C.

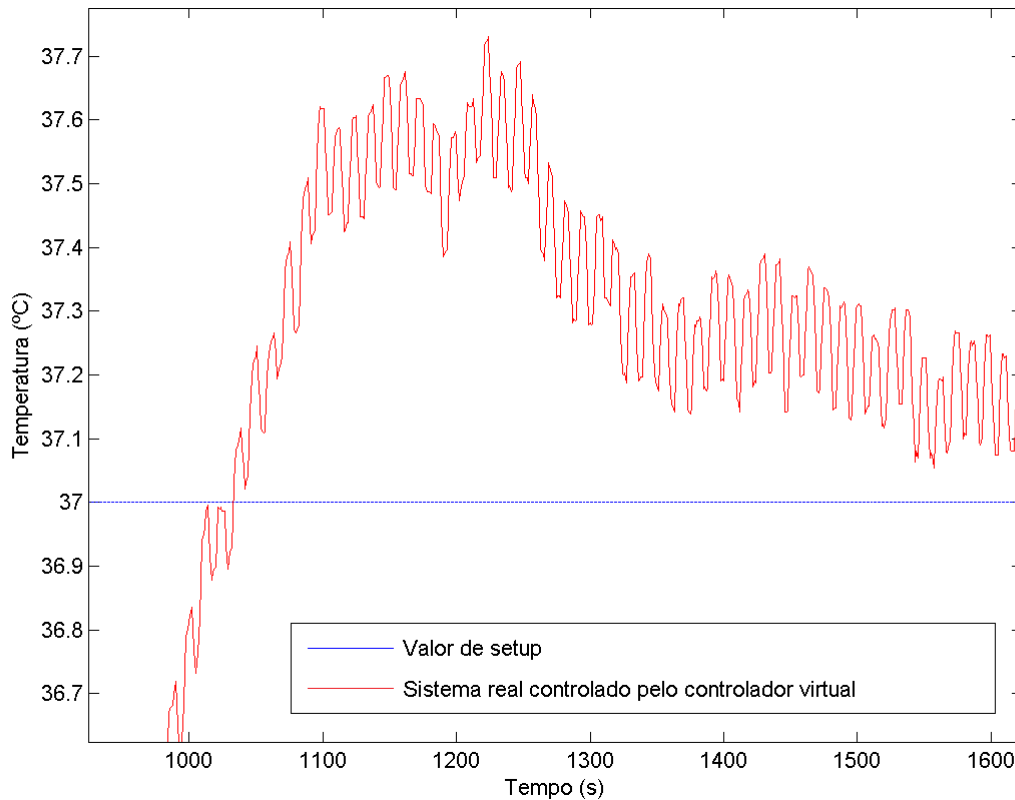


Figura 4.25: Teste do controlador virtual no sistema real com T_{setup} de 37°C , com maior foco na zona de estabilização.

4.13.3 Teste do controlador real no modelo virtual

Com a validação do controlador virtual no sistema real é finalmente possível efetuar a implementação do mesmo no controlador real. Assim, programou-se o *arduino* com as mesmas especificações que o controlador virtual e os mesmos parâmetros de PID. A melhor forma de testar se a programação é bem realizada é através de ensaios no sistema virtual, antes do sistema real, de forma a evitar possíveis erros e consequências materiais dos mesmos. Se o comportamento do sistema virtual for semelhante com ambos os controladores pode-se considerar que estes estão a funcionar da mesma forma, e que o *arduino* foi bem programado.

Para isso alterou-se o botão para a posição B, ligando o sistema virtual ao *arduino*. No que concerne à aquisição de dados utilizaram-se as funcionalidades do *ControlDesk*, tal como referido anteriormente. As condições inseridas no sistema modelado foram iguais às do ensaio da Secção 4.13.1. Para auxiliar na comparação entre os testes as Figuras 4.26 e 4.27 mostram a curva do controlo efetuado pelo controlador virtual, já testado anteriormente, e pelo *arduino* no modelo em *Simulink*.

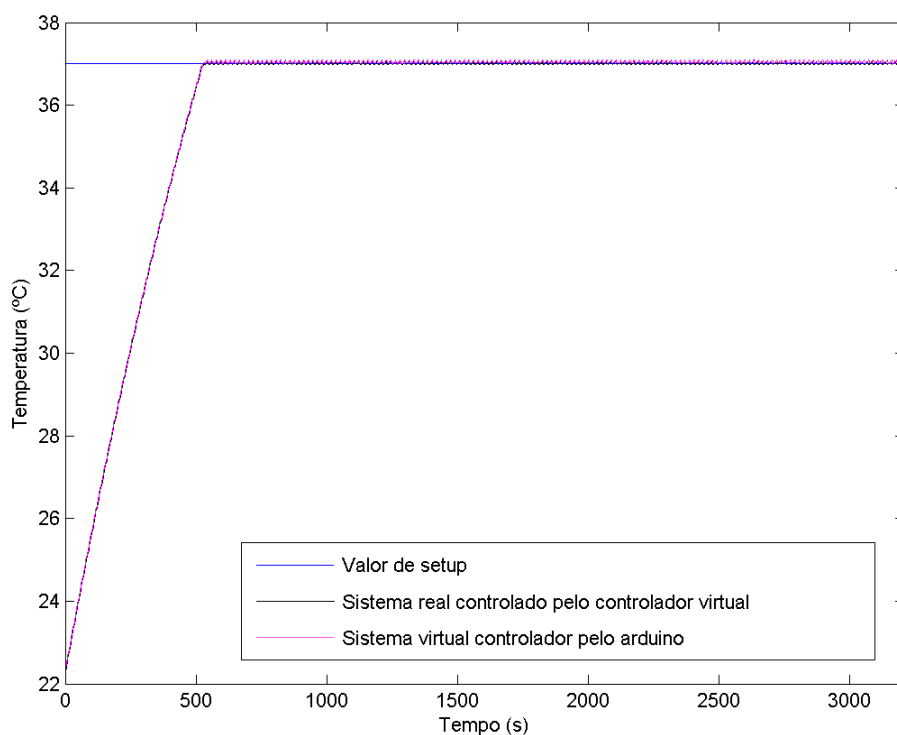


Figura 4.26: Teste do controlador virtual e do *arduino* no sistema virtual com T_{setup} de 37°C .

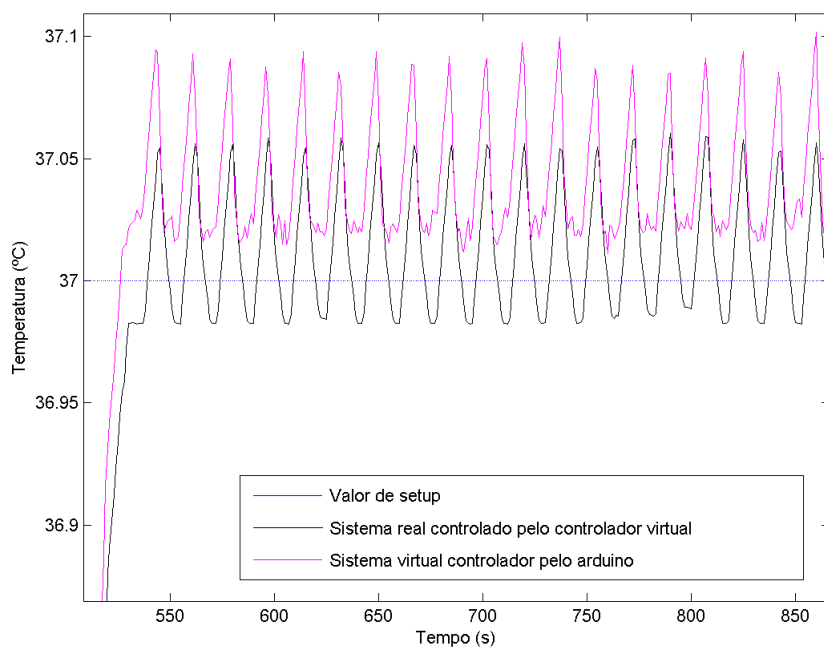


Figura 4.27: Teste do controlador virtual e do *arduino* no sistema virtual com T_{setup} de 37°C , com maior foco na zona de estabilização.

4.13.4 Teste do controlador real no modelo real

Após garantir que o *arduino* tem o comportamento esperado no controlo do sistema virtual, resta testá-lo no sistema físico. Esta representa a fase final do projeto, onde todos os componentes são reais e não existe qualquer tipo de modelo ou componente simulado. Poderá representar a etapa onde se valida o protótipo e o controlador real, antes da produção em série de um equipamento. Assim, alterou-se o botão de comutação para a posição A, ligando o *arduino* aos componentes físicos. A Figura 4.28 mostra a variação da temperatura da água no reservatório ao longo do tempo usando para aquisição de dados com o *ControlDesk*. Uma vez mais os valores de temperatura apresentados foram tratados por um filtro através do *MATLAB* para eliminar algum ruído.

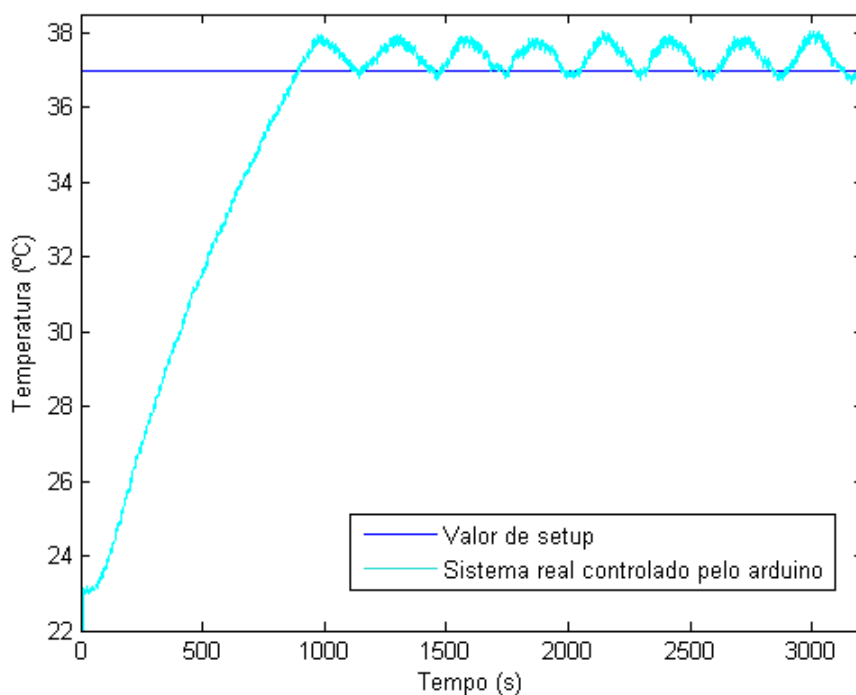


Figura 4.28: Teste do *arduino* a controlar o sistema real com Tsetup de 37°C.

4.13.5 Segundo teste dos controladores no modelo real

Anteriormente foram realizados ensaios de controlo do sistema real tanto pelo *arduino* como pelo *Simulink*. No entanto, tendo em conta que são vários os fatores que influenciam um sistema físico e que os primeiros testes foram realizados em dias diferentes, decidiu-se repetir os mesmos mas desta vez no mesmo dia e nas mesmas condições. Pretendeu-se com isto reduzir o número de variáveis que possam alterar os resultados nos testes, e tentar obter semelhanças entre os dois controladores. Seguiram-se exatamente os mesmos passos dos testes anteriores, deixando desta vez as mesmas condições tanto num como noutro controlador. A Figura 4.29 mostra a variação da temperatura da água no reservatório controlada nos 37°C, com as mesmas condições de controlo que os testes anteriores, tanto pelo *arduino* como pelo controlador virtual. Relembra-se que foi aplicado um filtro nos valores para retirar parte do ruído.

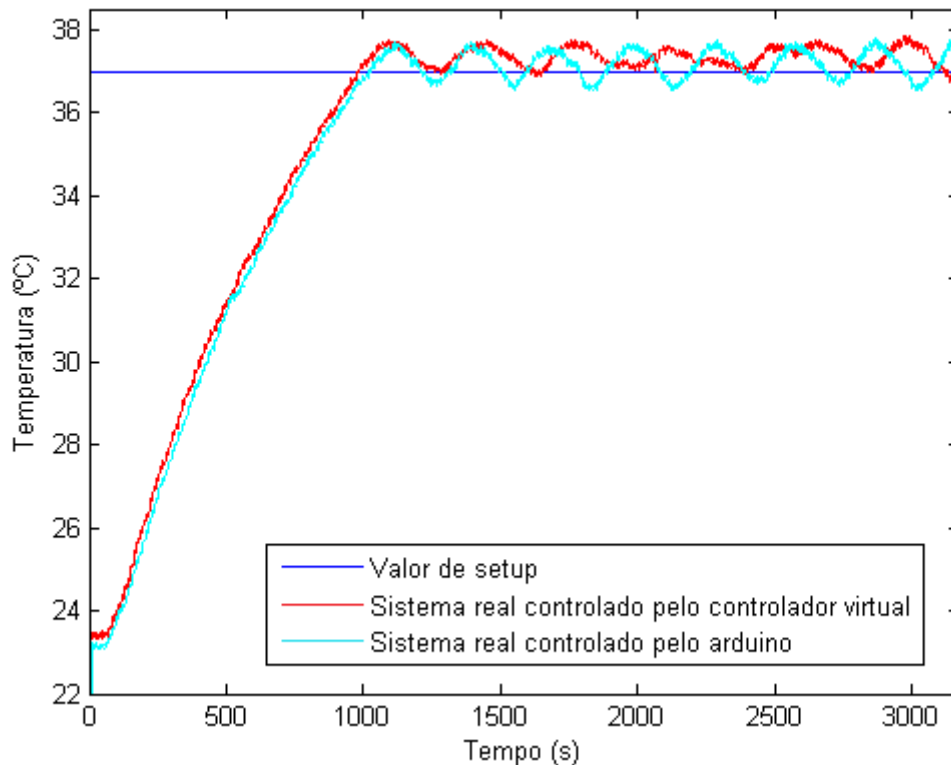


Figura 4.29: Teste do *arduino* e do controlador virtual no sistema real com Tsetup de 37°C.

4.14 Análise dos dados do caso de estudo 1

Analisando primeiramente o projeto do controlador, verificou-se que usando a metodologia proposta se conseguiu um controlador com uma boa capacidade de resposta para o sistema em questão. É de destacar a facilidade com que este conceito de bancada de ensaios permite projetar um controlador diretamente no sistema real, efetuando a aquisição dos sinais, simplesmente comutando os relés e mudando as opções de simulação no *Simulink*.

No que concerne à simulação do controlador virtual no sistema virtual (Secção 4.13.1) verificou-se que este responde como esperado, controlando o modelo simulado para a temperatura desejada e mantendo a temperatura com muito poucas variações.

Para o teste do controlador virtual no sistema real verificou-se que, apesar das variações, este respondeu bem e manteve a temperatura aproximadamente no valor pretendido. Logicamente que nesta parte se apresenta uma diferença de resposta comparando com o sistema virtual. Essa diferença pode dever-se ao facto do sistema estar a ser modelado de forma simples, não tendo em conta que a convecção no fluido demora o seu tempo e que a temperatura não é homogénea em todo o fluido no interior do reservatório.

No que diz respeito ao controlo do sistema virtual pelo *arduino* os resultados foram bastante satisfatórios, e observa-se que a componente de controlo foi implementada efi-

cazmente visto que quase se sobrepõe aos valores da Figura 4.22, que corresponde ao sistema virtual controlado pelo controlador virtual.

Nos testes efetuados pelo *arduino* no sistema real pode-se dizer que o sistema foi eficazmente controlado. No entanto o ideal seria o comportamento do sistema real ser semelhante tanto no controlo através do *arduino* como do *Simulink*. Uma das razões que podem justificar as diferenças observadas é o facto de os testes terem sido feitos em dias diferentes. Estes resultados vêm comprovar que é difícil obter valores semelhantes entre testes próximos, pelo facto de um sistema físico estar sempre em constante mudança e com vários fatores externos que provocam essas mesmas alterações.

De forma a despistar o problema descrito efetuou-se uma repetição dos testes no sistema real com ambos os controlos, virtual e *arduino*, desta vez no mesmo dia e seguidos. Verificou-se a esperada aproximação e a temperatura, além de estar bem controlada, comprovou-se que variava da mesma forma nos dois controlos. Comparando as duas curvas observa-se uma proximidade no comportamento, o que mostra que é possível a partir dos testes realizados com o controlador virtual no protótipo prever o comportamento do sistema e otimizar o controlador de forma mais segura. Mostrou-se ainda que o facto dos testes serem feitos em condições externas diferentes influencia os resultados de forma clara.

O caso de estudo correu com o sucesso pretendido. Verificou-se que é possível, com o mínimo de adaptações, construir uma plataforma de ensaios que permita quatro tipos de testes diferentes.

Mostrou-se ainda que essas quatro possibilidades são bastante úteis no projeto de controladores pois pode-se otimizar um controlador e testá-lo diretamente no sistema real através do *Simulink*. Este passo é vantajoso na medida em que só após testar o bom comportamento do controlador virtual no sistema real é que se parte para a programação do *arduino*, o que se apresenta como um avanço visto que não existe uma perda de tempo necessária para a programação de controladores reais sem a certeza que possam funcionar corretamente.

Capítulo 5

Caso de estudo 2

Com o do caso de estudo 1 verificou-se que o conceito de bancada de ensaios é exequível e, dessa forma, decidiu-se avançar o mais possível para um segundo caso de estudo. Neste, optou-se por implementar a ideia num esquentador da Bosh TT e efetuar todo o projeto elétrico. Devido às limitações de tempo e material, neste caso de estudo 2 apenas se efetuou o projeto elétrico e algumas das especificações chave para o colocar em prática. A ideia para este caso de estudo é que se realmente o conceito funciona com um número reduzido de atuadores e sensores também funcionará com sistemas mais complexos.

5.1 Definição do caso de estudo 2

O caso de estudo 1 serviu para sustentar o conceito proposto de bancada de ensaios, e que realmente este funciona. Com essa confirmação optou-se por avançar no trabalho e iniciar um segundo caso de estudo que desta vez envolva um equipamento mais complexo. Tendo em conta a parceria e o interesse da Bosch TT nesta ideia de bancada, foram fornecidas pelo colaborador da empresa as especificações de cada componente de um dos últimos esquentadores topo de gama, ainda em desenvolvimento. Essa escolha foi efetuada com base no interesse em aplicar o conceito de bancada ao mesmo.

Através da análise das características dos documentos técnicos foi possível delinear o condicionamento de sinal, para interação entre os diferentes sistemas e as adaptações necessárias ao *hardware* e *software* existente.

Vai-se considerar os mesmos moldes do caso de estudo 1, em que o *software* de simulação e aquisição de dados é o da dSPACE e o *software* de modelação é o *MATLAB/Simulink*. Assim, mantêm-se as três zonas de funcionamento importantes: o sistema real, constituído pelo esquentador, o controlador real e o sistema virtual.

O sistema real é um esquentador da *Bosch TT* onde apenas foram fornecidas as especificações e os requisitos elétricos dos componentes que o constituem. Por ser um modelo ainda em fase de protótipo, e para sua proteção, este não foi identificado nem se fará referência a qualquer outra informação do sistema que não seja relevante para a elaboração do projeto elétrico. No que diz respeito ao controlador real, este é projetado e programado pelos engenheiros da *Bosch TT*, interessando apenas os sinais de funcionamento. A nível da modelação do sistema e do controlador virtual também não se fará qualquer abordagem, exceto no que concerne às adaptações necessárias para o modelo receber e enviar sinais para o exterior.

Dos componentes que foram fornecidos consta a válvula de água com atuação por solenóide, a válvula *bypass* de água, a válvula de gás, o ventilador e os sensores NTC de temperatura. Cada um tem uma gama de funcionamento distinta, o que torna mais desafiante o projeto elétrico comparativamente ao caso de estudo 1. A Figura 5.1 mostra um possível esquema de montagem dos componentes referidos, para uma melhor percepção dos mesmos. Importa referir que o esquema apresentado é uma possível aproximação da realidade tendo em conta que com base nos dados fornecidos não é possível confirmar a disposição dos elementos no esquentador.

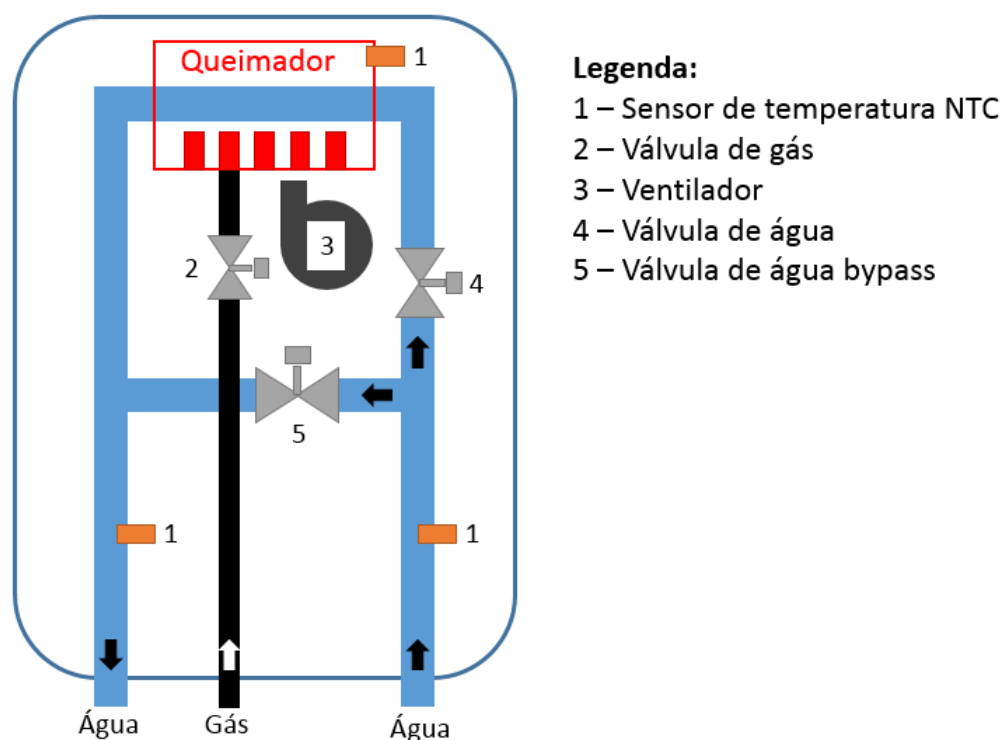


Figura 5.1: Esquema exemplificativo dos componentes distribuídos pelo esquentador

O princípio de funcionamento envolve contactos para comutar entre simulações, e os respetivos condicionadores de sinal para possibilitar a comunicação entre os diferentes sistemas. O esquema da Figura 5.2 mostra, de forma simplificada, como será projetada a interface. Onde "C1", "C2" e "C3" representam os contactos para ligar os pares de sistemas, "CC" os contactos junto à carta de aquisição de dados para a troca de sinais para os diferentes tipos de interação a que o sistema virtual está sujeito e, por último, os "M" simbolizam os módulos para condicionamento do sinal.

Conforme se pode verificar, pretende-se que o caso de estudo 2 em tudo se assemelhe ao caso de estudo 1 no que diz respeito aos meios usados, às técnicas de condicionamento de sinal e à distribuição dos elementos na bancada. A ideia é desenvolver os módulos de interface entre o sistema real e os restantes elementos de aquisição e controlo.

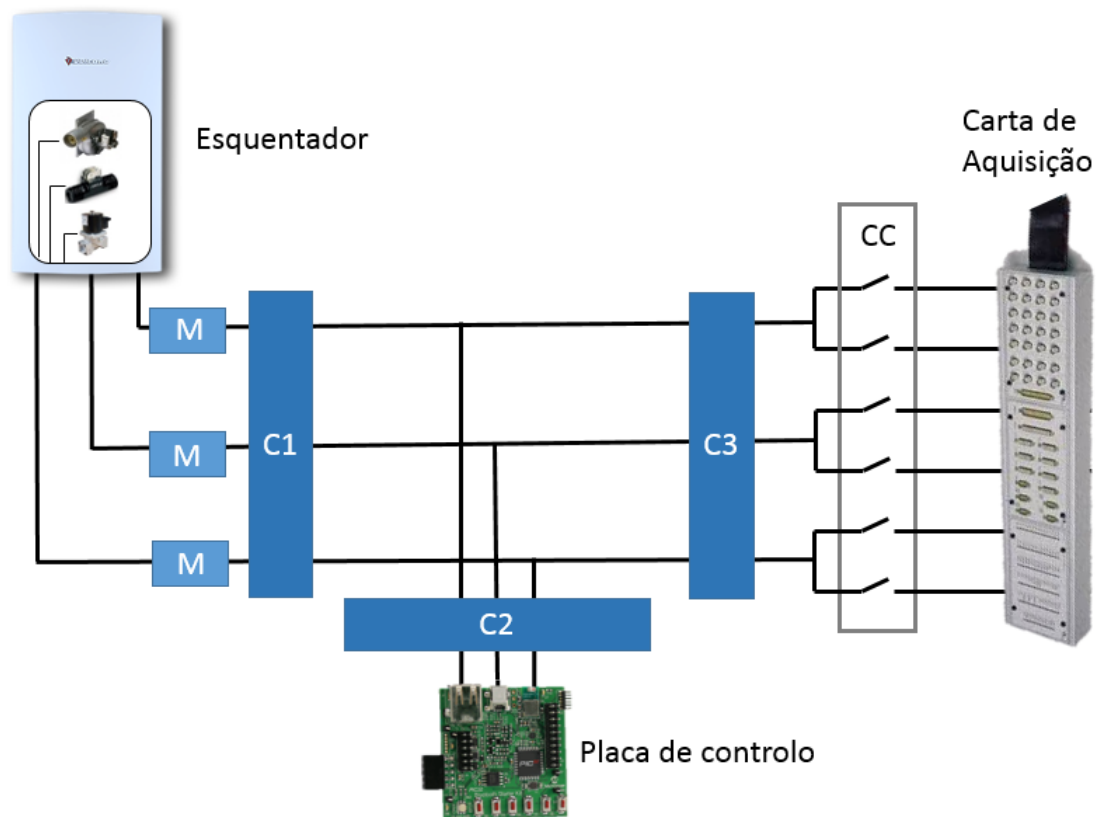


Figura 5.2: Esquema demonstrativo das ligações pretendidas para o caso de estudo 2

5.2 Condicionamento de sinal

Tal como já foi abordado no primeiro caso de estudo, para o bom e rápido funcionamento deste tipo de bancada de ensaios é preciso efetuar o condicionamento de sinal necessário para a interação entre os sinais da carta, do microcontrolador e dos componentes. Para isso foram fornecidos pela Bosch TT os dados referentes a um esquentador no que diz respeito ao sinais de funcionamento, para possibilitar o projeto elétrico. Contudo, as especificações fornecidas são referentes a ordens de compra, pelo que não estão totalmente completas devido ao esquentador ainda estar em desenvolvimento. Então, para as especificações em falta, foram consideradas algumas gamas de funcionamento com base no que se encontra no mercado, estando todas as especificações resumidas na Tabela 5.1.

Com base nas especificações fornecidas e nos requisitos de sinais da carta de aquisição de dados, relembrando que para sinais analógicos é de 0 a 10 VDC e digitais de 5 VDC, efetuou-se o projeto de cada módulo de condicionamento de sinal.

Importa referir que apesar dos módulos de condicionamento entre o microcontrolador e os componentes do esquentador já estarem à priori desenvolvidos pela Bosch TT, como não se sabe o conteúdo de cada um estes vão ser igualmente abordados tal como no caso de estudo anterior. Assim, como não foi fornecido outro tipo de informação, optou-se por considerar a gama de funcionamento típica de um microcontrolador, com sinais digitais de 5 VDC e analógicos de 0 a 5 VDC.

Tabela 5.1: Resumo do levantamento dos requisitos elétricos de cada componente

Componente	Características Técnicas	
Válvula bypass de água	Alimentação	24 VDC
	Controlo	5 VDC
Válvula de água	Controlo do motor	24 VDC
	Sensor de caudal	5 - 18 VDC
Sensor de temperatura NTC	Alimentação	5 VDC
	Saída	0 - 5 VDC
Válvula de gás	Alimentação	24 VDC
	Controlo	5 VDC
Ventilador	Alimentação	7 - 45 VDC
	Sensor de velocidade	12 VDC
Carta de aquisição de dados	Entradas e saídas analógicas	0-10 VDC
	Entradas e saídas digitais	5 VDC
Microcontrolador	Entradas e saídas analógicas	0-5 VDC
	Entradas e saídas digitais	5 VDC

5.2.1 Válvula bypass

Um dos componentes existente no esquentador é uma válvula *bypass* de água. Esta funciona a 24 VDC, e com corrente entre 200 e 500 mA. O funcionamento é efetuado através de um sinal PWM que define a abertura da válvula. As especificações não são totalmente claras, e considerou-se uma tensão de controlo de 5 VDC, tipicamente a PWM, e uma tensão de alimentação a 24 VDC.

Tanto na carta de aquisição de dados como no microcontrolador o sinal digital PWM de 0 a 5 VDC. Para este acionar a válvula é necessário um módulo de interface que permita o seu controlo. Para isso usou-se um relé para a alimentação direta a 24 VDC, quando se comuta para o sistema real, e um circuito de interface para envio do PWM de forma a não danificar as saídas digitais, conforme se observa na Figura 5.3.

No que concerne à opção de simulação, onde o microcontrolador controla diretamente o sistema virtual, o sinal PWM pode ser enviado diretamente sem qualquer interface pelo meio visto que a gama é a mesma. É claro que nesta situação apenas é necessário enviar a tensão de controlo para o modelo.

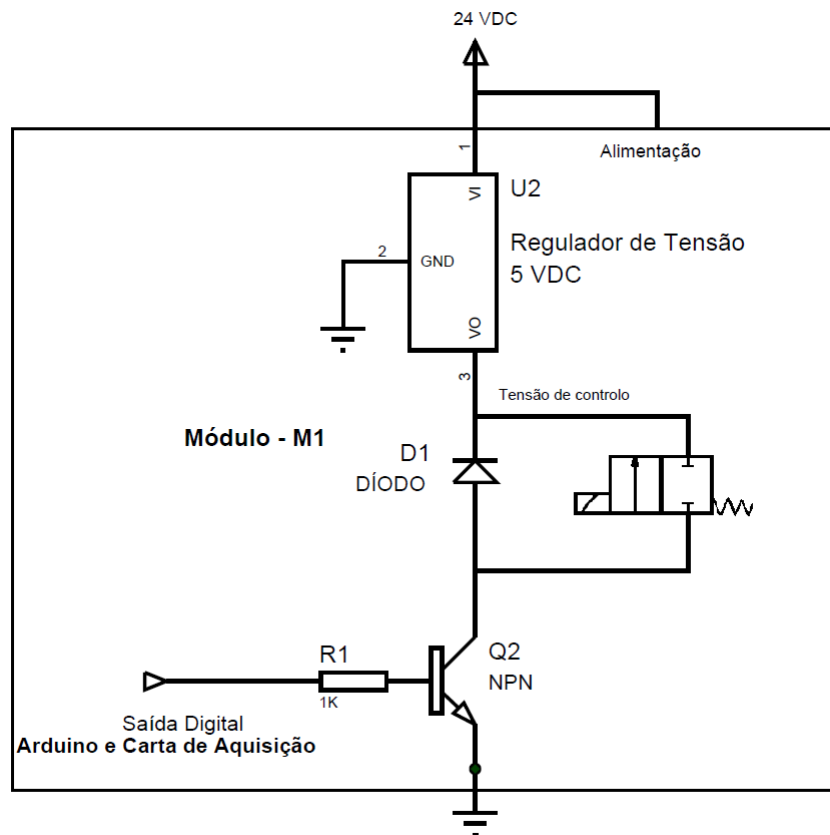


Figura 5.3: Módulo M1 com o condicionamento entre a válvula *bypass* e os controladores real e virtual

5.2.2 Válvula de água

No que diz respeito à válvula de água esta não é comum, e tem associados vários componentes. Tem acoplado um motor de passo que funciona na gama dos 24 VDC. Além disso, é equipada com um sensor de caudal que, caso seja atuado, emite um sinal com tensão de funcionamento normal entre 5 e 18 VDC e corrente máxima de 15 mA. Ainda mede a temperatura através de um sensor NTC, de resistência variável. Apesar de estar tudo acoplado à mesma válvula, os módulos de condicionamento para cada um devem ser feitos em separado devido à gama de sinais distinta.

No que concerne ao motor de passo da válvula, este é unipolar de 4 fases. Basicamente estes motores convertem pulsos elétricos em movimento mecânico, gerando assim a rotação. Pretende-se que através de quatro sinais digitais se gere a rotação do mesmo, e para isso é necessário um driver de interface, como é o caso do L293. Este é, no fundo, uma ponte H num circuito integrado, que fornece os drivers necessários para cargas indutivas, como os motores de passo, no máximo até 1 A de corrente e de 4.5 a 36 VDC de tensão. Além dos sinais para gerar movimento, o motor ainda tem mais quatro ligações que se descrevem na Figura 5.4. Representam-se pelos números de 1 a 5 as ligações para gerar rotação e o ponto comum dos 24 VDC, e pelos números 6 e 7 os sinais de controlo para dar o sentido de rotação ao motor. Estes últimos, pela falta de informação vão-se

considerar como dois sinais digitais que indicam para que lado roda o motor. Finalmente o número 8 corresponde à terra do sistema do motor.

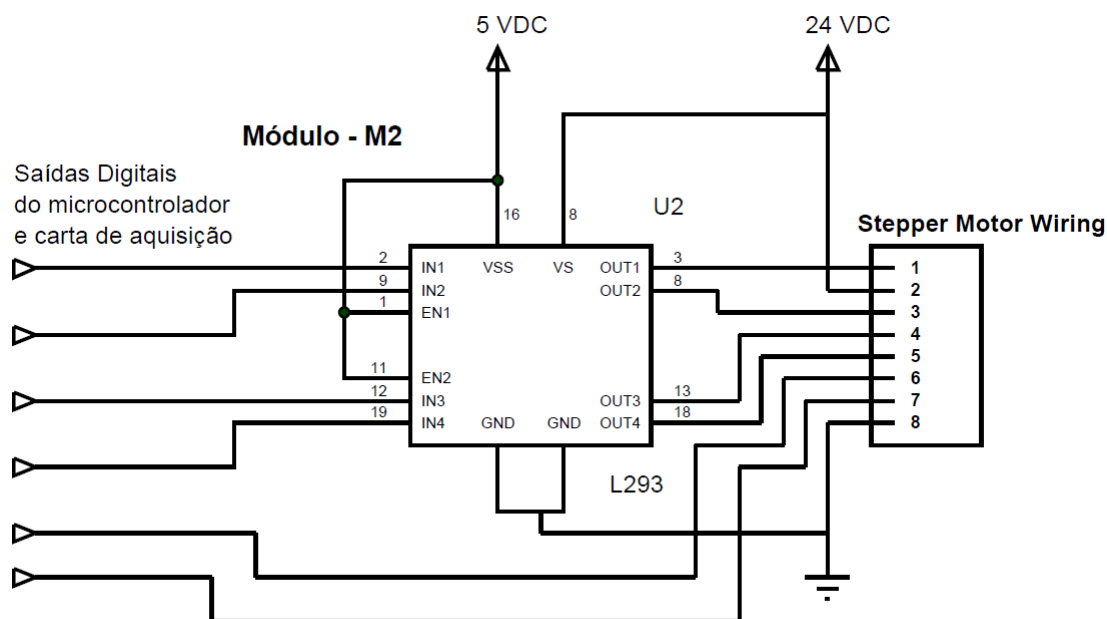


Figura 5.4: Módulo M2 com o condicionamento relativo ao sinal do motor da válvula de água

Outro grupo de ligações existente nessa válvula é do sensor de caudal. Este funciona através do princípio de Hall, onde a circulação do fluido na válvula provoca a rotação de uma turbina que emite pulsos numa determinada gama de tensões. Como interface apenas se aplicou um limitador de tensão de 5 VDC, para se ler nas entradas digitais do *hardware* de controlo e instrumentação (Figura 5.5). A interpretação da velocidade a que fluido passa é efetuada por *software*, através da interpretação do sinal referido.

A nível de ligação entre o microcontrolador, os sensores e o *driver* do motor abordado não é necessário qualquer tipo de interface, visto que o sinal é enviado e recebido na gama dos 5 VDC entre ambos.

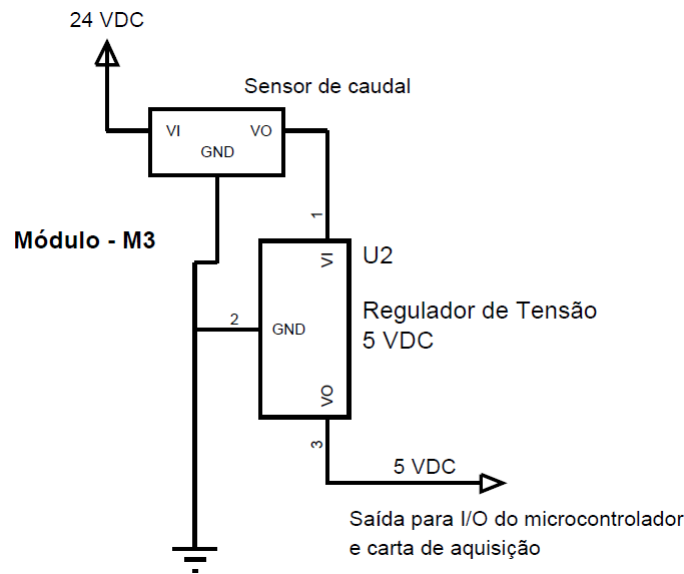


Figura 5.5: Módulo M3 com o condicionamento relativo ao sensor de caudal

5.2.3 Sensor de Temperatura NTC

Os sensores de temperatura de resistência variável, tal como o próprio nome indica, têm como princípio de funcionamento a variação da sua resistência elétrica à medida que a temperatura varia, em que o valor dessa resistência equivale a uma determinada temperatura. O sensor NTC é um desses exemplos, onde quanto maior a resistência elétrica menor a temperatura.

Uma das formas pensada para determinar a resistência do NTC foi projetada através de um divisor de tensão simples. Usando a resistência do NTC e outra resistência conhecida, sabendo a tensão de entrada e medindo a tensão de saída consegue-se chegar ao valor do NTC (Figura 5.6). No entanto, esta montagem poderia trazer problemas de ruído e de incerteza na medição da tensão de saída, podendo o valor calculado para o NTC ter vários erros associados

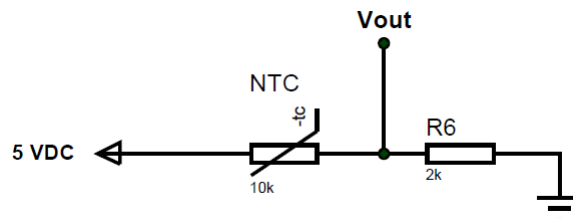


Figura 5.6: Primeiro módulo desenvolvido para interface do sensor de temperatura NTC

Como alternativa ao divisor de tensão pensou-se num *driver* de corrente, representado na Figura 5.7, que possibilita saber a resistência do sensor de temperatura com base na corrente que passa por ele e na tensão. O amplificador usado permite filtrar algum ruído, e no sensor tem-se uma passagem de corrente fixa.

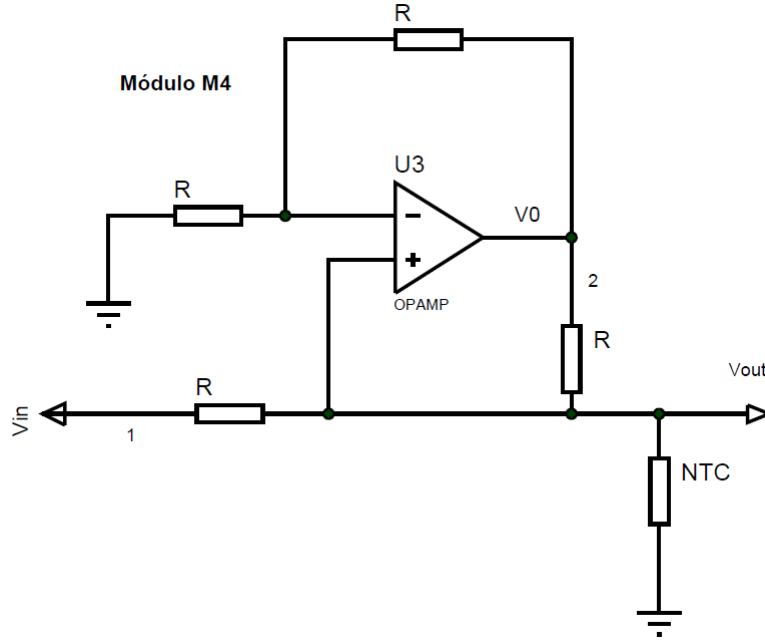


Figura 5.7: Módulo M4 com o condicionamento relativo ao sinal do sensor de temperatura

Através da configuração apresentada sabe-se que a corrente que passa no sensor NTC é dada por:

$$I_{NTC} = I_1 + I_2 \quad (5.1)$$

Como as resistências "R" têm todas o mesmo valor e a tensão V2 é metade da tensão "Vout" tem-se que:

$$I_{NTC} = \frac{V_{in} - \frac{V_{out}}{2}}{R} + \frac{V_{out} - \frac{V_{out}}{2}}{R} = \frac{V_{in}}{R} \quad (5.2)$$

Como a tensão "Vin" é conhecida e a resistência "R" também, consegue-se achar o valor da corrente que passa no sensor de temperatura. Assim, medindo "Vout" no sistema de aquisição é possível saber o valor da resistência do NTC através do quociente entre essa tensão e a corrente.

Para dimensionar as resistências "R" e saber qual o valor de tensão "Vin" a aplicar considerou-se uma gama de trabalho entre 0 °C e 100 °C, que tem como valor de resistência do NTC associado de 23,73 kΩ e 0,8041 kΩ, respetivamente. Se o objetivo passar por ter uma tensão de aquisição máxima de 5 VDC, implica que a corrente máxima do NTC seja de aproximadamente 0.006 A. Então, de forma a obter um valor inferior definiu-se que a resistência "R" seria de 1kΩ e a tensão "Vin" de 5 VDC.

A nível das ligações analógicas entre o microcontrolador e a carta de aquisição e controlo considerou-se a gama de trabalho de 0 a 5 VDC, de forma a reduzir módulos de condicionamento, e tem-se que modelar o sinal de temperatura da mesma forma que este é enviado após a passagem pelo módulo M4.

5.2.4 Válvula de gás

A nível de válvula de gás, trata-se de uma válvula de solenóide alimentada a 24 VDC, sendo regularmente utilizada na maior parte dos esquentadores da *Bosch TT*.

Esta é controlada através de um sinal PWM com amplitude a 5 VDC, e para isso tem de ser enviado o sinal digital da carta de aquisição e do microcontrolador para a colocar em funcionamento. Para interface de sinal usou-se o mesmo circuito de interface da válvula bypass representado pela Figura 5.3, tendo em conta as semelhanças de funcionamento.

5.2.5 Ventilador

Existem esquentadores que são equipados por ventiladores, nomeadamente aqueles que podem funcionar em locais de difícil arejamento (tiragem forçada de gases). Este dispositivo ajuda a tornar a chama estável e no controlo de emissões de gases.

No caso deste esquentador específico, o motor do ventilador funciona à tensão de 45 VDC. A sua velocidade é modulada através de PAM (*Pulse Amplitude Modulation*). Para melhor se entender o conceito, enquanto que o PWM modula a largura de pulso e mantém a amplitude, o PAM modula a amplitude de pulso. A amplitude do pulso pode ir de 7 a 45 VDC, e com intensidade de corrente de 1,7 A, tendo proteção até 3,2 A.

Tendo em conta que é necessário enviar um sinal de elevada frequência mas com amplitude variável, não se pode recorrer a um relé de estado sólido, tal como se usou em componentes anteriores para PWM. Assim, é necessário enviar um sinal analógico de tensão variável de 0 a 5 VDC a partir dos controladores, que consiga modelar uma tensão de 7 a 45 VDC. Para tal escolheu-se um amplificador operacional não inversor de potência de ganho 10, sendo que a tensão de saída no microcontrolador tem que ser modelada de 0,7 a 4,5 VDC para se obter a amplitude desejada após o *opamp*. No que diz respeito à carta de aquisição manteve-se o mesmo circuito de interface, apesar de permitir uma maior amplitude nas saídas analógicas. Não se considera relevante esse fator, e pode ser perfeitamente definida a amplitude de saída igual à do microcontrolador, para poupança de módulos.

A Equação 5.3 apresenta os cálculos efetuados para as resistências do *opamp* e a Figura 5.8 o circuito projetado.

$$Ganho = \frac{V_{out}}{V_{in}} = \frac{R1 + R2}{R2} \Leftrightarrow 10 = \frac{1000 + R2}{R2} \Leftrightarrow R2 \simeq 110\Omega \quad (5.3)$$

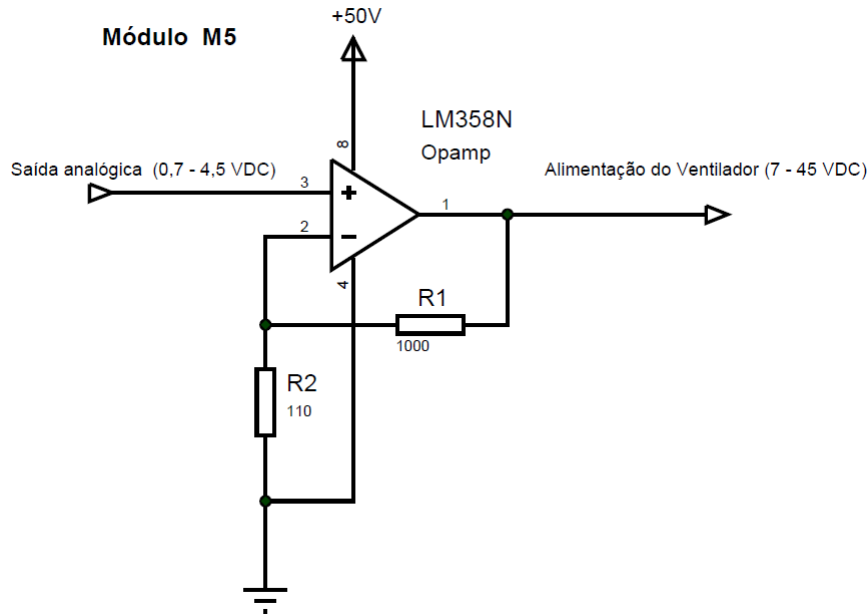


Figura 5.8: Módulo 5 - Circuito de interface para o sinal PAM de acionamento do ventilador

Para o controlo da velocidade do ventilador, este está equipado com um sensor de Hall, que envia pulsos que o sistema de controlo interpreta como rotação. No equipamento em análise uma volta corresponde a 4 impulsos. Para isso, o ventilador tem um circuito integrado que no fim envia pulsos com uma amplitude de 12 VDC. Para detetar, no microcontrolador ou carta de aquisição, basta ativar uma entrada digital a 5 VDC, pelo que um simples divisor de tensão é suficiente para o efeito. Observe-se a Equação 5.4 com os cálculos das resistências para converter os 12 VDC em 5 VDC, e a Figura 5.9 com o circuito aplicado. Importa referir que se poderia recorrer à alternativa usada no circuito do sensor de caudal da Figura 5.5.

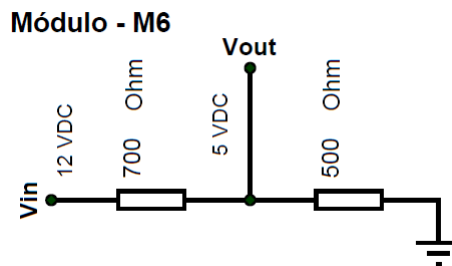


Figura 5.9: Módulo 6 - Circuito de interface para o sinal do sensor de Hall do ventilador

$$V_{out} = \frac{R_2}{R_1 + R_2} \times V_{in} \Leftrightarrow 5 = \frac{500}{R_1 + 500} \times 12 \Leftrightarrow R_1 = 700\Omega \quad (5.4)$$

5.3 Comutação entre simulações

Tal como no caso de estudo 1, pretende-se elaborar o plano para o caso de estudo 2 que facilite igualmente a comutação entre simulações de forma rápida e simples.

Vão-se manter os mesmos moldes do caso 1, com as 4 possibilidades de comutação e os 3 contactos entre sistemas, através de relés.

Além dos contactos entre cada um dos sistemas, há também outros que estão implementados junto à carta de aquisição e controlo. Assim, consegue-se comutar entre controlo do sistema virtual por microcontrolador e controlo do sistema real por controlador virtual sem a necessidade de alterar as ligações dos cabos. Observe-se o exemplo da Figura 5.10.

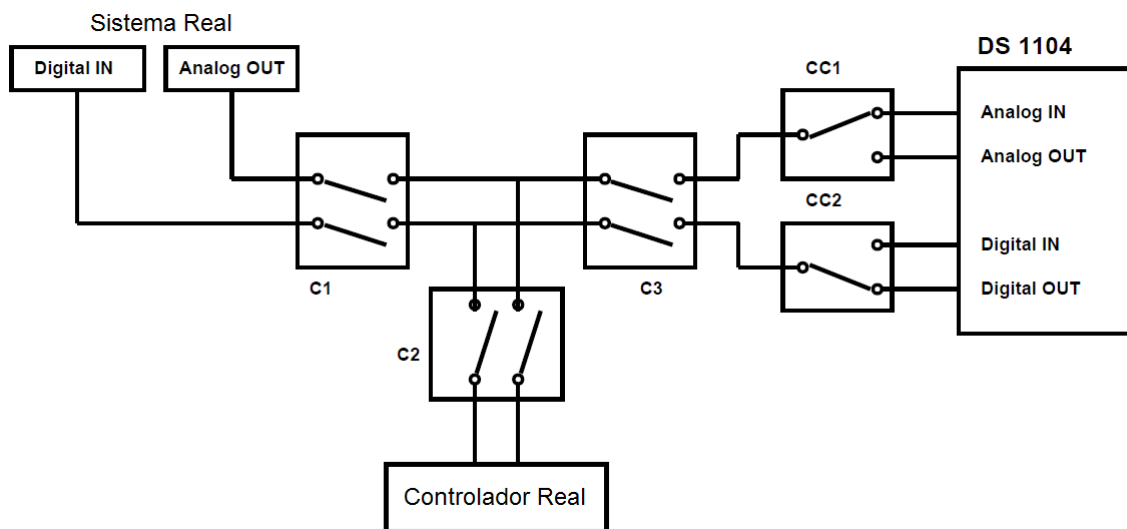


Figura 5.10: Exemplo de comutação para 2 sinais distintos

No exemplo representado têm-se 2 sinais no sistema real, um analógico a ser enviado e outro digital a ser recebido. No que concerne à simulação quando se liga o sistema virtual ao real pretende-se que se fechem os contactos "C1" e "C2", e que o sinal analógico se ligue ao "Analog IN" e o digital ao "Digital Out" da DS1104. Quando o objetivo é ligar o sistema virtual ao microcontrolador, este tem que ser conectado ao "Analog OUT" e ao "Digital IN". Neste caso, o sinal analógico poderia ser de temperatura enquanto que o digital um PWM de controlo.

As implementações dos contactos "CC1" e "CC2" foram aplicadas junto à carta de aquisição pelo facto do sistema virtual ter o papel de simulação e de controlo. Já o microcontrolador não necessita deste tipo de cuidados, visto que apenas tem a função de controlo e, portanto, os sinais que envia e recebe são sempre os mesmos.

No que concerne às ligações entre os sistemas deixam-se em aberto duas possibilidades de comutação, assemelhando-se uma delas ao caso de estudo 1. Pode-se efetuar a comutação com um botão mecânico para acionar os relés referentes, e na componente de simulação deixar a opção para alternar entre simulações internamente. A outra possibilidade é apenas se comutar no modelo virtual, ou seja ao correr a simulação em tempo real haver um comando com opção de simulação em que se envia um sinal digital, dependendo do que se pretende, para o exterior comutando-se os relés pretendidos. Ambas são

válidas; no entanto, o ideal seria haver as duas opções por segurança, caso ocorra algum problema.

De forma a aproximar o presente caso de estudo do anterior, vai-se deixar a hipótese de seleção manual no projeto elétrico realizado.

O Apêndice D mostra o projeto elétrico das ligações existentes com os módulos de condicionamento e os contactos para comutação entre simulações. Nesse projeto deixaram-se os sinais da temperatura constantemente ligados nos diferentes sistemas, tal como se fez no caso de estudo 1.

5.4 Adaptações no *Simulink*

Para que seja possível implementar as ideias de rápida troca de simulação é necessário adaptar o *Simulink* para o efeito. Vão-se referir as ideias chave para que se consiga, de forma simples, trabalhar com sinais externos ou internos.

A ideia principal a reter é o facto de o sistema modelado trabalhar com variáveis de controlo. São essas que têm que estar preparadas para rápido envio para o exterior ou receção, dependendo do sinal em si. Assim sendo, devem-se definir variáveis gerais, comuns a todo o modelo para que este facilmente se adapte aos vários tipos de simulação através de, por exemplo, "Multi-Port Switch". Tal como se fez para o caso de estudo 1, com esse bloco é possível seleccionar o tipo de simulação pretendida e, assim, uma variável pode assumir diferentes valores, reais ou simulados.

O *Simulink*, a par dos pacotes de interface com a dSPACE, tem blocos associados para efetuar o envio e receção dos sinais analógicos e digitais. Esse processo já foi mencionado na Secção 4.6, pelo que não será necessário voltar a explicar o funcionamento dos mesmos. Ainda assim, vai-se referir o exemplo de um sinal correspondente a um componente, sabendo que os restantes sinais seguem o mesmo princípio.

5.4.1 Exemplo de adaptação

Para melhor se entender a forma fácil como se pode adaptar um modelo em *Simulink* para trabalhar com diferentes tipos de simulação escolheu-se um componente e o respetivo sinal, que neste caso foi o sinal PWM enviado para a válvula bypass, por exemplo .

Esse sinal tem que ser gerado no *Simulink* para enviar ao sistema virtual e para o sistema real; além disso, deve estar preparado para receber o sinal externo do microcontrolador para aplicar ao sistema modelado. Assim, pode-se utilizar o princípio representado na Figura 5.11, onde a constante "Tipo de simulação" vai assumir valores de 1 a 3, consoante o tipo de simulação, e passá-los ao bloco "Simul". Se esse bloco assumir o valor 1, o sinal PWM da válvula ("PWM_valv") recebe o PWM simulado; caso seja o valor 2, o PWM passa a ser o gerado no microcontrolador assumindo esse valor para o modelo virtual; se for 3 o sinal é aquele que será enviado para o sistema real.

Esse bloco "PWM_valv" é então ligado ao sistema da válvula para gerar a abertura e o fecho da mesma através de um PWM simulado, caso se escolha a opção 1; à entrada digital da carta (Figura 5.12), caso receba o PWM vindo do microcontrolador fazendo com que o sistema modelado da válvula receba o PWM real caso se opte pela opção 2; à saída digital da carta (Figura 5.13) se o objetivo for enviar o sinal PWM simulado diretamente para a válvula real, efetuando o controlo da mesma.

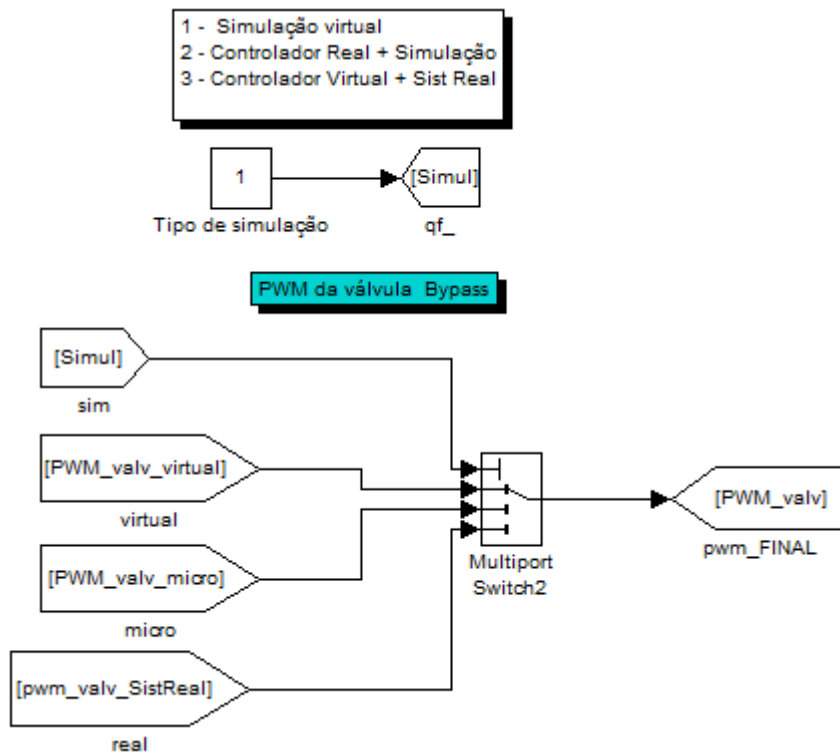


Figura 5.11: Exemplo de comutação do sinal PWM da válvula *bypass*

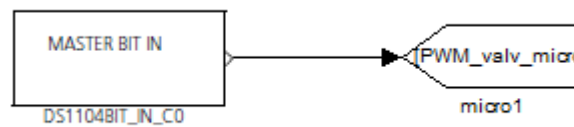


Figura 5.12: Recepção do sinal PWM vindo do microcontrolador



Figura 5.13: Envio do sinal PWM para o componente real

5.5 Procedimento experimental

Esta caso de estudo 2, devido à inexistência de material, apenas vai ficar na fase de projeto, não havendo qualquer teste. Ainda assim, aborda-se um possível procedimento experimental que terá em conta aqueles que podem ser os passos de desenvolvimento do esquentador na *Bosch TT*.

A linha cronológica que pode ser seguida para se efetuar uma possível componente ex-

perimental pode ser observada na Figura 5.14. O processo assemelha-se ao procedimento aplicado no caso de estudo 1.

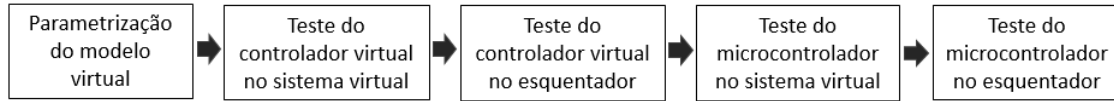


Figura 5.14: Possível fluxo experimental para o caso de estudo 2

5.5.1 Modelo virtual

O primeiro passo passa pela modelação do sistema virtual em *MATLAB/Simulink*. Aqui desenvolvem-se os modelos dinâmicos através da implementação das equações do sistema, e projetam-se os respetivos controladores.

Com o sistema virtual pode-se observar o seu comportamento e afinar alguns parâmetros em ambiente simulado, bem como testar alguns casos extremos de comportamento, protegendo assim os equipamentos em caso de erro humano.

É importante que se apliquem as técnicas para interação com sinais externos e se modelem os sinais a enviar e a receber, para posteriormente serem usados noutra tipo de simulação.

5.5.2 Interação entre modelo real e virtual

Comutando os contactos para possibilitar a interação entre o sistema real e o virtual, é possível controlar o primeiro através do controlador virtual, ou até mesmo alternar entre componentes reais e simulados.

Esta interação traz várias vantagens, como o caso de possibilitar a parametrização do modelo virtual, aproximando-o ao comportamento do sistema físico para melhor projeto dos controladores.

Além disso, usando o controlador virtual para controlar o esquentador, pode-se otimizar o mesmo verificando o comportamento do sistema real e testando vários parâmetros de controlo. No fim, pretende-se que o controlador virtual esteja conforme pretendido ao controlar o sistema real.

5.5.3 Interação entre a placa de controlo e o modelo virtual

Após se parametrizar o modelo virtual e otimizar o controlador no esquentador é possível passar, com maior confiança, para uma outra fase do desenvolvimento, que passa pela programação da placa de controlo. Pretende-se que o controlo implementado na placa seja em tudo semelhante ao do projetado no modelo em *Simulink*.

Com a placa de controlo programada, pode-se mudar a opção de simulação para a ligar ao sistema virtual. Assim, é possível controlar o modelo virtual usando um controlador real, ou seja, aplicar a técnica de simulação com *hardware-in-the-loop*. Tendo em conta que o modelo foi parametrizado, e já se aproxima ao comportamento do esquentador real, é possível testar se o controlador real está bem programado e se o comportamento do sistema se aproxima das simulações anteriores.

Esta opção possibilita anular alguns erros na programação do microcontrolador, evitando assim um possível incidente caso esta fosse logo testada no esquentador.

5.5.4 Interação entre placa de controlo e protótipo

Com a placa de controlo aprovada, através dos dados do modelo virtual, é possível passar para a última fase de teste onde apenas estão envolvidos componentes reais.

Tal como no caso de estudo 1, nesta última parte pode-se verificar o comportamento do esquentador, sistema real, quando controlado pela placa de controlo, controlador real, e verificar se todo o processo está em correto funcionamento.

5.6 Considerações sobre o caso de estudo 2

Não há grandes conclusões a retirar deste caso de estudo pelo facto de não existir uma componente prática passível de ser analisada. No entanto, pode-se considerar que ficaram algumas etapas base para a aplicação do conceito de bancada de ensaios que se pretende.

Se forem realmente realizados os módulos de condicionamento necessários, e se aplicarem os blocos corretos no *Simulink*, é possível obter uma bancada que permita efetuar o tipo de simulações mencionadas no caso de estudo 1.

É importante ainda relembrar que o projeto elétrico referenciado foi uma aproximação aos dados que foram fornecidos pelos colaboradores da *Bosch TT*, visto que o esquentador em análise tem alguns dos seus componentes ainda a ser desenvolvidos e, por isso, não tem os requisitos elétricos completamente definidos.

Tal como no caso de estudo 1, é impreterível que o modelo virtual envie e receba os sinais do exterior da mesma forma que o microcontrolador, para que as alterações que se façam no projeto do controlador sejam válidas e se aproximem da melhor forma ao controlador real que será posteriormente programado.

Capítulo 6

Conclusões

A exigência cada vez maior do consumidor e a necessidade de fazer cada vez melhores produtos, e de forma mais rápida, obriga a que se invista ainda mais no setor de testes da indústria de desenvolvimento.

É extremamente importante o uso de modelos dinâmicos pois permite prever o comportamento de sistemas, o que se apresenta como uma vantagem no que concerne acima de tudo à segurança, poupança de meios e investimento. Apesar do tempo usado inicialmente a modelar o sistema de forma eficaz, é possível que este apresente vantagens a longo prazo pelas razões mencionadas.

Olhando para as bancadas de ensaios do mercado, principalmente as de testes de sistemas térmicos, observa-se uma grande limitação no que diz respeito à flexibilidade e adaptação a vários tipos de simulação. A maior parte permite o teste apenas numa fase final do projeto, o que aumenta a insegurança do produto numa fase inicial do desenvolvimento. Encontram-se algumas bancadas que permitem *hardware-in-the-loop*, mas não há referências a sistemas térmicos.

Através do conceito de bancada de ensaios proposto passa a ser possível auxiliar o processo de desenvolvimento numa fase bem inicial, onde se consegue testar os controladores diretamente no protótipo. Isto apresenta-se como uma enorme vantagem, visto que antes mesmo de se construir e programar a placa de controlo pode-se já ter um modelo de controlo testado e aprovado. Além disso, o teste de microcontroladores no sistema virtual traz alguma segurança, sendo possível detetar defeitos de programação e até mesmo monitorização num sistema simulado, sem qualquer risco associado. Por último, ainda possibilita testar o protótipo na fase final, com o controlador já programado e com o sistema real perfeitamente construído, assemelhando-se assim à maior parte das bancadas de ensaio com esta última, e geralmente única, opção.

Foi possível demonstrar, através do caso de estudo 1, que o conceito de *layout* proposto pode funcionar desde que se façam as adaptações no *Simulink* e se considerarem os módulos de condicionamento de sinal necessários para os sistemas interagirem entre eles. Existem alguns problemas de ruído devido ao material utilizado não se adequar na totalidade, o que é perfeitamente normal. Ainda assim, consideram-se os resultados obtidos bastante satisfatórios, conseguindo-se uma otimização do controlador através do modo 2 de operação, ou seja, o controlador virtual a controlar o reservatório de aquecimento.

No que diz respeito aos testes efetuados no caso de estudo 1, para validar o conceito de bancada, verificou-se que é possível aproximar o controlador virtual ao real desde que os parâmetros de controlo estejam muito próximos. A prova disso está nos testes

realizados por ambos os controladores no modelo virtual (Secção 4.13.1 e 4.13.3), onde houve grande aproximação das curvas de temperatura.

Observando-se o teste realizado ao sistema real, quer com o controlador virtual quer com o *arduino* (Secção 4.13.5), sustentou-se ainda mais a ideia de que é possível, na fase de projeto do controlador no *Simulink*, prever o comportamento que o sistema terá aquando o funcionamento através do controlador real, dado que as curvas em ambos os controladores estão próximas.

Conseguiu-se com este trabalho desenvolver um conceito de bancada de ensaios que consegue superar algumas das limitações das bancadas existentes, e disponibilize uma gama mais ampla de tipos de testes, destacando-se o *hardware-in-the-loop-simulation* em sistemas térmicos.

No caso de estudo 2 deixaram-se algumas das ideias principais para a construção da mesma, e alguns cuidados necessários para ser possível implementar as várias formas de teste. O objetivo foi fundamentar a ideia de que projetando os módulos de interface adequados é possível aplicar o conceito apresentado a qualquer sistema, mais simples ou mais complexo. Obviamente que a complexidade do sistema também aumenta o trabalho de adaptação dos modelos e do sinal.

Apesar da componente prática deste trabalho envolver um pequeno número de atuadores e sensores, o que é mais importante salientar é que é possível aplicar o conceito de bancada proposto permitindo, de entre outras possibilidades, *hardware-in-the-loop-simulation*.

As principais considerações a ter em conta é que é possível desenvolver esta bancada de ensaios desde que se projetem os módulos de interface necessários entre os diferentes sinais dos componentes. Além disso, deve existir a preocupação para o modelo ser adaptado para conseguir reproduzir os sinais que vêm dos outros equipamentos, e interpretá-los. Também se deve aproximar o controlador virtual ao controlador real da melhor forma possível, para se obterem melhores resultados.

O objetivo foi alcançado, conseguindo-se desenvolver um conceito de bancada de ensaios com os requisitos mencionados, validando-o através de um caso de estudo e de sucessivas experimentações laboratoriais.

6.1 Sugestões de Trabalhos futuros

O conceito de bancada de ensaios foi comprovado através de um caso de estudo simples, e não houve a possibilidade de avançar para um projeto de grande dimensão. Nesse sentido, uma proposta de trabalho futuro seria continuar o caso de estudo 2 na componente prática, construindo assim uma bancada de ensaios, equipada com um esquentador e todos os restantes componentes de interface, controlo e monitorização. Além disso, implementar o condicionamento de sinal abordado e validar o bom funcionamento da interface.

Do ponto de vista de validação dos resultados seria importante, após se atualizar uma bancada à escala industrial, aplicar as alterações à mesma e testar um equipamento diferente para garantir que o conceito é exequível seja para que sistema for.

Além da hipótese de avançar para uma bancada à escala industrial, também seria interessante estudar a influência do *hardware* de simulação e aquisição de dados neste tipo de testes. Neste trabalho recorreu-se à dSPACE, e nesse sentido seria interessante fazer os mesmos testes com *software* e *hardware* de outro fabricante, por exemplo da

National Instruments.

Bibliografia

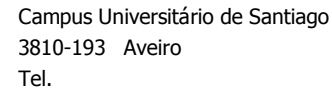
- [1] J. A. Ledin, “Hardware-in-the-Loop Simulation,” *Embedded Systems Programming*, pp. 42–60, 1999.
- [2] C. D. d. S. Barros, “Desenvolvimento de Plataformas de Automação Digitais,” Dissertação de Mestrado, 2013.
- [3] H. K. Fathy, Z. S. Filipi, J. Hagena, e J. L. Stein, “Review of hardware-in-the-loop simulation and its prospects in the automotive area,” vol. 6228, pp. 62 280E–62 280E–20, 2006. [Online]. Disponível: <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=1286270>
- [4] R. Isermann, J. Schaffnit, e S. Sinsel, “Hardware-in-the-loop simulation for the design and testing of engine-control systems,” *Control Engineering Practice*, vol. 7, n° 5, pp. 643–653, 1999. [Online]. Disponível: <http://www.sciencedirect.com/science/article/pii/S0967066198002056>
- [5] J. Ferreira, “Modelação de Sistemas Hidráulicos para Simulação com Hardware-in-the-loop,” Tese de Doutorado, Universidade de Aveiro, 2003.
- [6] C. Köhler, *Enhancing Embedded Systems Simulation*, 2011. [Online]. Disponível: <http://www.springerlink.com/index/10.1007/978-3-8348-9916-3>
- [7] Simulink Coder, “Real-Time Workshop,” 2012. [Online]. Disponível: <http://www.mathworks.com/products/simulink-coder/index.html>
- [8] K.-S. Hong, H.-C. Sohn, e J. K. Hedrick, “Modified Skyhook Control of Semi-Active Suspensions: A New Model, Gain Scheduling, and Hardware-in-the-Loop Tuning,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 124, n° 1, p. 158, 2002.
- [9] M. B. Design, M. B. D. Mbd, T. Dc, P. C. Target, e P. C. Targetbox, “xPC Target,” pp. 2–5.
- [10] P. Mosterman, S. Prabhu, e A. Dowd, “Embedded real-time control via MATLAB, Simulink, and xPC target,” pp. 1–28, 2005. [Online]. Disponível: http://link.springer.com/chapter/10.1007/0-8176-4404-0_{_}18
- [11] A. Martin e M. R. Emami, “An architecture for robotic hardware-in-the-loop simulation,” *2006 IEEE International Conference on Mechatronics and Automation, ICMA 2006*, vol. 2006, pp. 2162–2167, 2006.
- [12] W. River, “VxWorks Application Programmer’s Guide, 6.2.”

- [13] H. M. Silva, “Simulação com Hardware In the loop aplicada a veículos submarinos,” Dissertação de Mestrado, 2008.
- [14] F. Use e R.-t. Workshop, “Real-Time Windows Target For Use with Real-Time Workshop.”
- [15] D. Xue e Y. Chen, *System Simulation Techniques with MATLAB and Simulink*, 2013, vol. First Edit.
- [16] E. T. Enikov e G. Campa, “Mechatronic Aeropendulum : Demonstration of Linear and Nonlinear Feedback Control Principles,” *Ieee Transactions on Education*, vol. 55, n° 4, pp. 538–545, 2012.
- [17] Opal-RT Technologies Inc, “RT-LAB Quickstart Guide,” 2010.
- [18] S. Abourida e C. Dufour, “Real-time PC-based simulator of electric systems and drives,” *Applied Power Electronics Conference and Exposition, 2002. APEC 2002. Seventeenth Annual IEEE*, vol. 1, n° c, pp. 433–438 vol.1, 2002. [Online]. Disponível: http://ieeexplore.ieee.org/ielx5/7757/21321/00989281.pdf?tp=989281&isnumber=21321&delimiter=026E30F&http://ieeexplore.ieee.org/xpls/abs/_all.jsp?arnumber=989281
- [19] S. Gmbh, “dSPACE Catalog 2016,” p. 56, 2016.
- [20] R. M. B. R. Pilão, “Estudo do Comportamento Térmico de Esquentadores Domésticos a Gás,” Dissertação de Mestrado, Instituto Superior de Engenharia do Porto, p. 108, 1996.
- [21] P. M. P. Semblano, “Desenvolvimento de Bancas de Ensaio,” Relatório de Estágio, Faculdade de Engenharia da Universidade do Porto, 2006.
- [22] N. Gu e J. M. Ni, “Simulation of engine cooling system based on AMESim,” *2009 2nd International Conference on Information and Computing Science, ICIC 2009*, vol. 4, n° 2, pp. 117–120, 2009.
- [23] A. V. Loesch, A. dos Santos, e M. E. Garcia, “Bancada didática para testes de falhas de sistemas de refrigeração,” Paraná, 2012. [Online]. Disponível: <http://repositorio.roca.utfpr.edu.br/jspui/handle/1/831>
- [24] K. K. Gunda, “Adjustable Speed Drives Laboratory Based on Dspace Controller,” Dissertação de Mestrado, Faculty of the Louisiana State University and Agricultural and Mechanical College, 2008.
- [25] J. Bursill, “Evaluation of Heat Pump Water Heater Performance and Optimization Strategies In Cold Climates,” Dissertação de Mestrado, University of British Columbia, p. 148, 2015.
- [26] S. Devasia, S. Member, E. Eleftheriou, S. O. R. Moheimani, e S. Member, “A Survey of Control Issues in Nanopositioning,” *IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY*, vol. 15, n° 5, pp. 802–823, 2007.

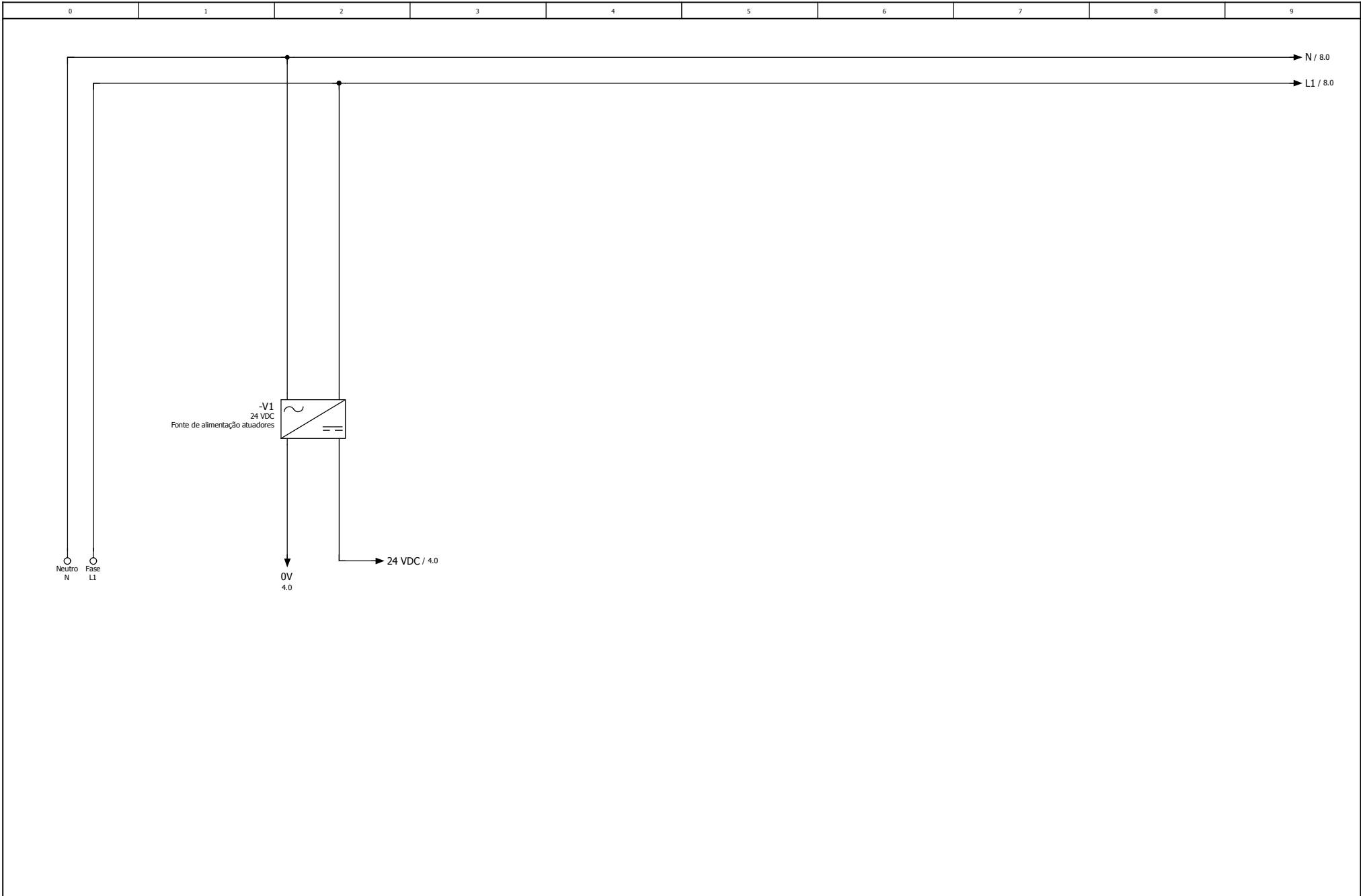
Anexos

Apêndice A

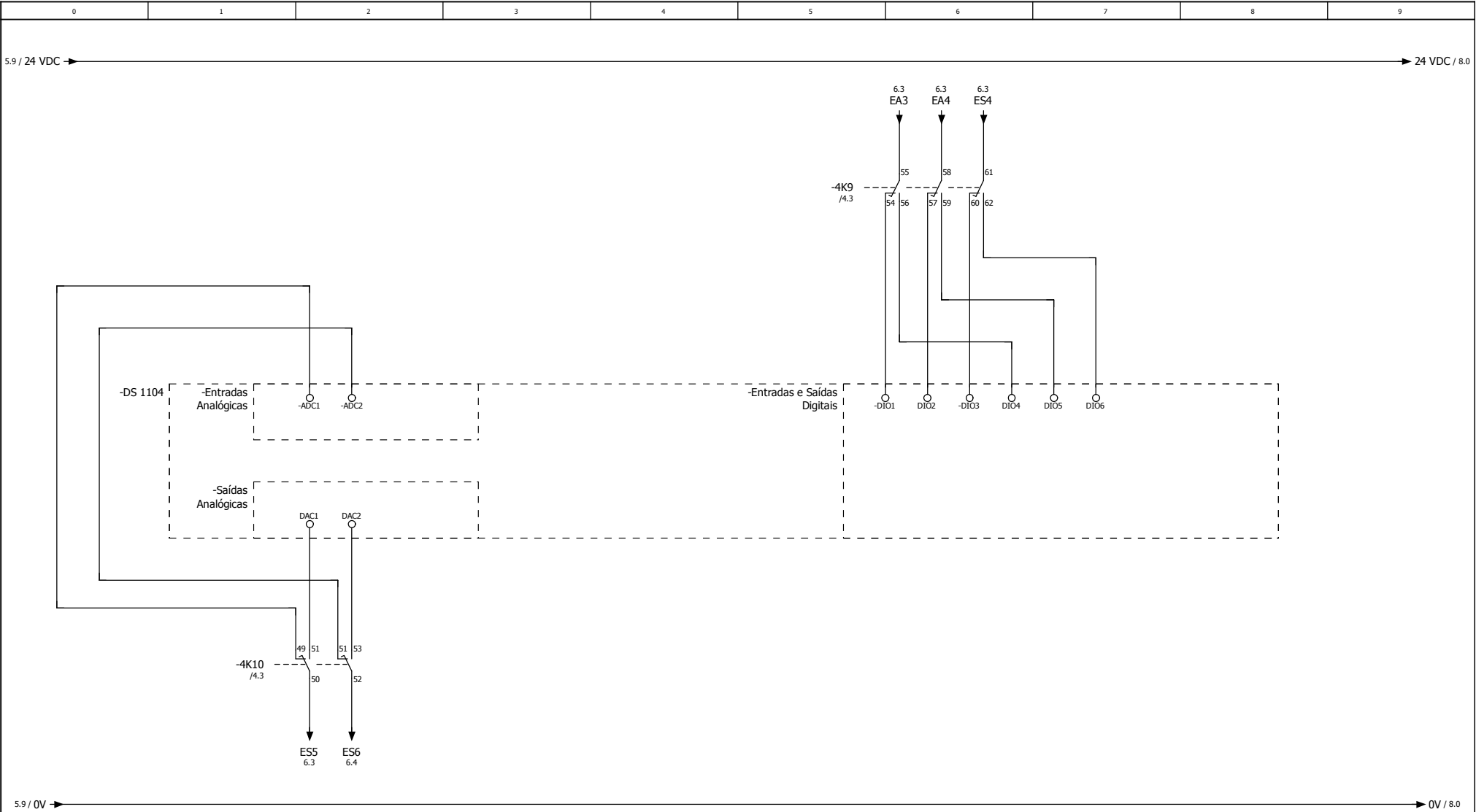
Projeto elétrico do caso de estudo 1

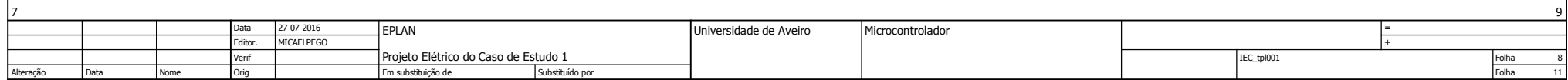


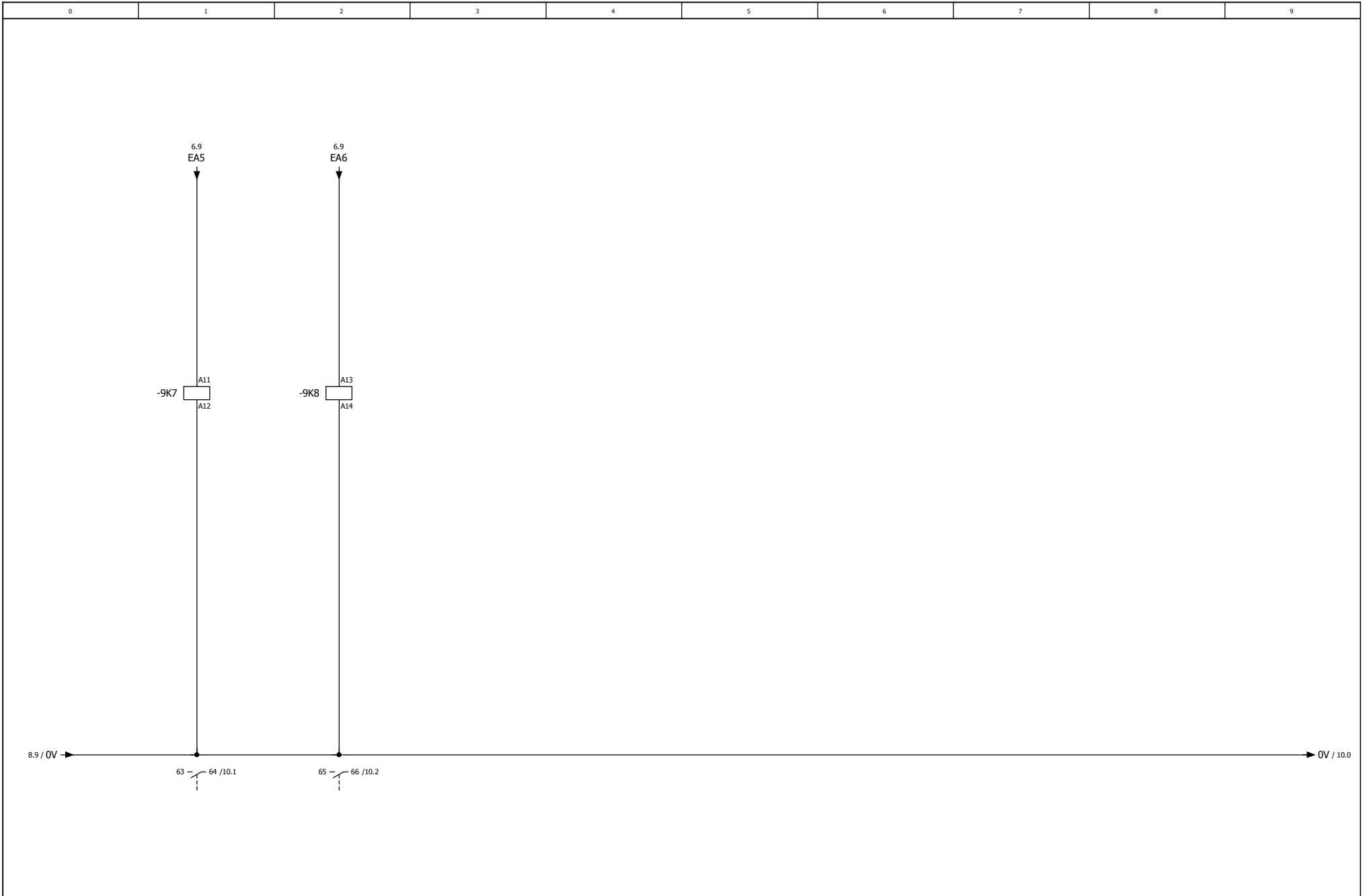
Empresa / Cliente	Projeto Elétrico do Caso de Estudo 1		
Descrição do projecto	IEC_tpl001		
Número do desenho	EPLAN		
Comissão			
Fabricante (Empresa)	Universidade de Aveiro		
Coluna	Projeto Elétrico do Caso de Estudo 1		
Nome do projecto	Projeto Bancada de Ensaios		
Realizar			
Tipo			
Localização de instalação			
Responsável pelo projecto			
Características da peça			
Criado em	16-03-2016		
Editado em	27-07-2016	de (abreviatura) MICHAELPEGO	Quantidade de páginas 11



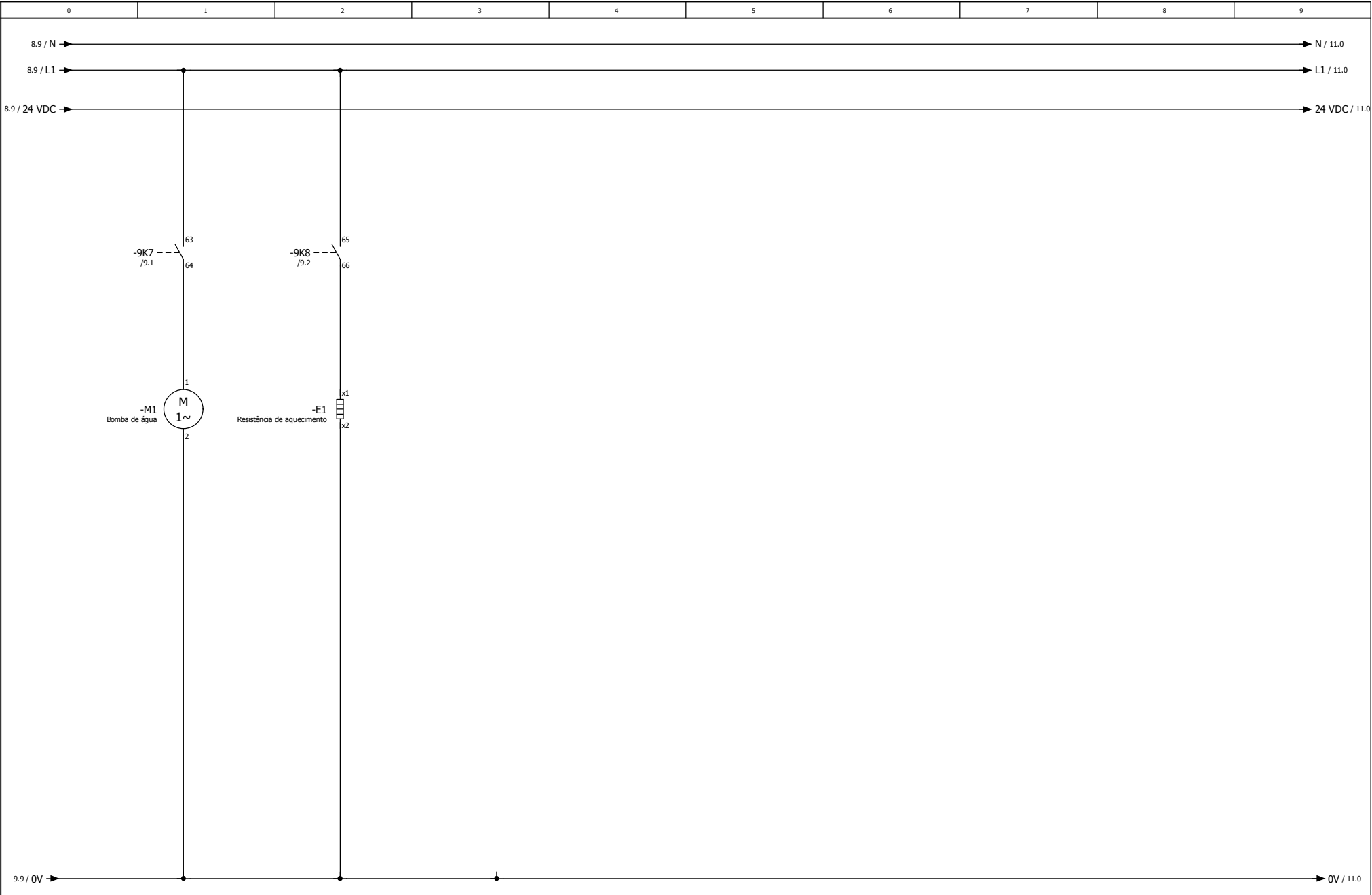
1												4	
			Data	27-07-2016	EPLAN		Universidade de Aveiro	Alimentação			=		
			Editor.	MICHAELPEGO							+		
			Verif		Projeto Elétrico do Caso de Estudo 1								
Alteração	Data	Nome	Orig		Em substituição de	Substituído por				IEC_tpl001	Folha	3	
									Folha	11			



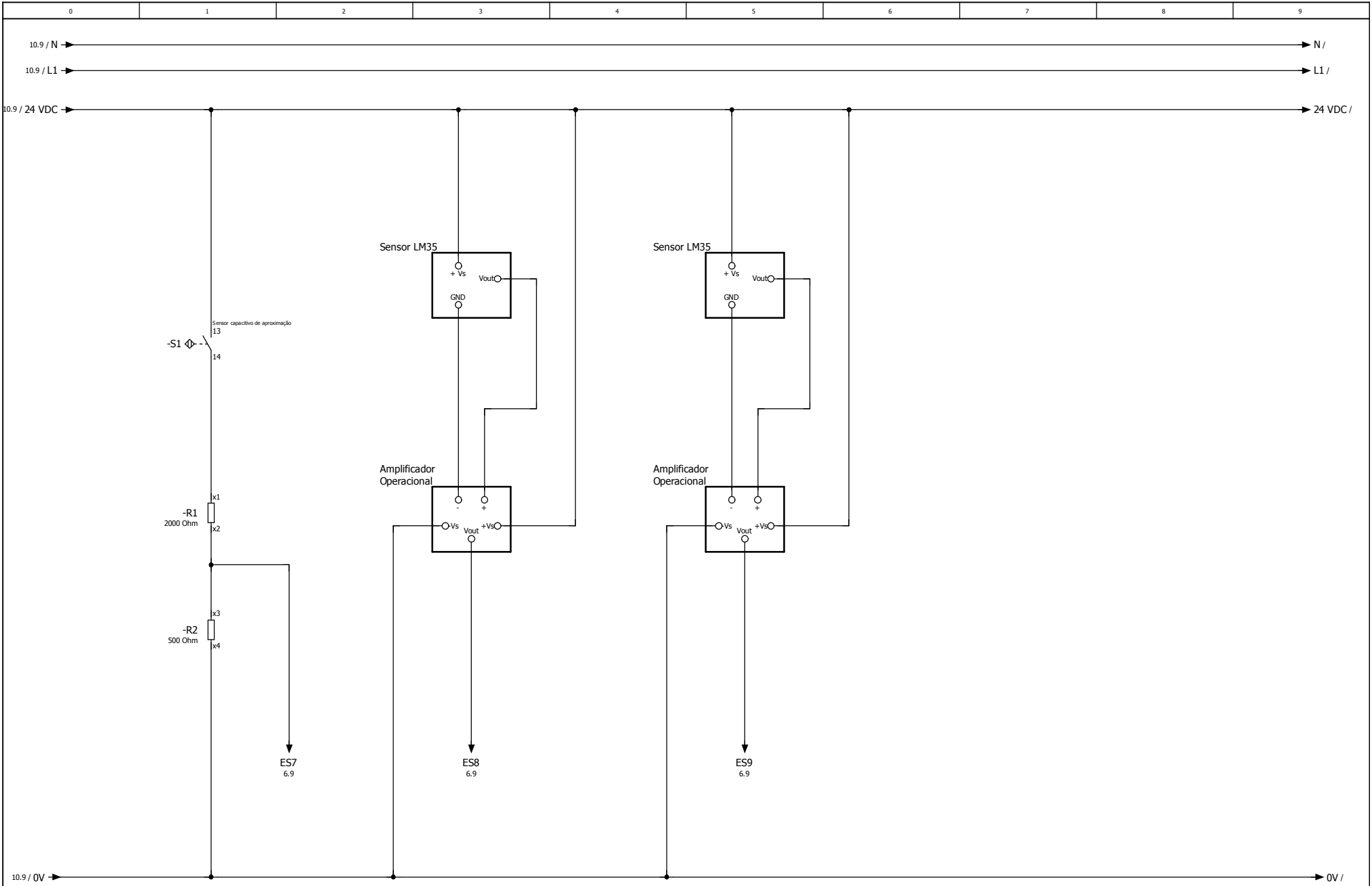




8											10										
			Data	27-07-2016	EPLAN			Universidade de Aveiro		Interface do circuito						=					
			Editor.	MICAELPEGO	Projeto Elétrico do Caso de Estudo 1											+					
			Verif																		
Alteração	Data	Nome	Orig		Em substituição de		Substituído por									IEC_tpl001			Folha 9		
															Folha 11						



9		11									
			Data	27-07-2016	EPLAN	Universidade de Aveiro	Atuação Sistema			=	
			Editor.	MICAELPEGO						+	
			Verif		Projeto Elétrico do Caso de Estudo 1						
Alteração	Data	Nome	Orig		Em substituição de			IEC_tpl001		Folha	10
					Substituído por					Folha	11



Apêndice B

Código implementado no arduino

```

#include <PID_v1.h>
#include "TimerOne.h"

//=====PINOS=====//
#define resist 10
#define bomba 7
#define snivel 2
//=====ESTADO INICIAL=====//
int xx = 0;
float Tsetup = 37;
float rho = 1000; //ró da água
float cp = 4186; // cp da agua
float Tamb = 23.4;

float PotMax = 2000;
float qin, qout, qentrada = 0.86; //l/min
float Tin, TIR, Tres, TRR;
float Pteorica, Pefetiva, Pfinal;
int i;
float Ttotal = 65536;
int count = 0;
float conv = 0.9;
float R = 0.1316;

float h = 0.14; //altura

float qout_an;
float qoutMAX = 0.5; // L/min

float duty = 0;

//----PID TUNING PARAMETERS-----//

float kp = 8.73; //Initial Proportional Gain
float ki = 216.2; //Initial Integral Gain
float kd = 0; //Initial Differential Gain
//float Setpoint, Input, Output; //These are just variables
for storing values.

double Setpoint, Input, Output, In, Set; //These are just

```

```

variables for storing values
PID myPID(&Input, &Output, &Setpoint, kp, ki, kd, DIRECT);

const int sampleRate = 1; // Variable that determines how
fast our PID loop

// Communication setup
const long serialPing = 500; //This determines how often we
ping our loop
// Serial pingback interval in milliseconds
unsigned long now = 0; //This variable is used to keep track
of time
// placeholder for current timestamp
unsigned long lastMessage = 0; //This keeps track of when our
loop last

// last message timestamp.
int freq;
float tempo2;
//float errSum, lastErr;
//unsigned long lastTime;
void setup()
{

  Serial.begin(9600);
  pinMode(snivel, INPUT);
  pinMode(bomba, OUTPUT);
  pinMode(resist, OUTPUT);
  //===== pwm =====//
  freq = 20;
  Ttotal = 1000000 / freq;
  Timer1.initialize(Ttotal);          // initialize timer1,
and set a 1/2 second period
  //Timer1.pwm(9, (duty/100)*1023);    // setup
pwm on pin 9, 50% duty cycle
  Timer1.attachInterrupt(callback); // attaches callback()
as a timer overflow interrupt

  myPID.SetMode(AUTOMATIC); //Turn on the PID loop
  myPID.SetSampleTime(sampleRate); //Sets the sample rate

```

```

Serial.println("Begin"); // Hello World!
lastMessage = millis(); // timestamp

}

void loop()
{
  //===== TEMPERATURA ENTRADA =====//
  Tin = 0;
  for (i = 0; i <= 5; i++) // gets 7 samples of temperature
  {
    TIR = (float(analogRead(A0)) * 5 / (1023)) / 0.1;
    Tin = Tin + TIR;
    // delay(100);
  }
  Tin = Tin / 6.0;
  Tin = 22.5;
  //===== TEMPERATURA SAIDA/RESERVATORIO =====//
  Tres = 0;
  for (i = 0; i <= 5; i++) // gets 8 samples of temperature
  {
    TRR = (float(analogRead(A1)) * 5 / (1023)) / 0.1;
    Tres = Tres + TRR;
    // delay(100);
  }
  Tres = (Tres / 6.0);
  Tres = Tres + 0.4;
  //===== CAUDAL SAIDA(SIMULADO) =====//

  // qout_an = (float(analogRead(A2)) * (5.0 / 1023.0)) *
(qoutMAX / 5);
  qout_an = 0.34;
  // qout_an=0;
  //===== SENSOR DE NIVEL =====//
  if (digitalRead(snivel) == HIGH) //se nivel=max bomba
desliga.
  {
    digitalWrite(bomba, LOW);
    qin = 0;
  }
}

```

```

    }
else
{
    digitalWrite(bomba, HIGH);
    qin = qentrada;
}
qout = 0.34; //L/min

//===== MODELAÇÃO POTENCIA =====//
Pteorica = (cp / 60) * (qout * Tsetup - qin * Tin) + (5 *
(Tsetup - Tamb) / (R / h)) + (conv * (Tsetup - Tamb));
Pefetiva = (cp / 60) * (qout * Tres - qin * Tin) + (5 *
(Tres - Tamb) / (R / h)) + (conv * (Tres - Tamb));

if (Pteorica <= 0)
{
    Pteorica = 0;
}
if (Pefetiva <= 0)
{
    Pefetiva = 0;
}

Set = Pteorica * 255 / 2000;
In = Pefetiva * 255 / 2000;

if (Set < In) //SALVAGUARDAR O FACTO DO PID NAO FUNCIONAR
COM VALORES NEGATIVOS
{
    Setpoint = Pefetiva * 255 / 2000;
    Input = Pteorica * 255 / 2000;
}
else
{
    Setpoint = Pteorica * 255 / 2000;
    Input = Pefetiva * 255 / 2000;
}
myPID.Compute();

```

```

now = millis();

Output = Output * 2000 / 255;
Input = Input * 2000 / 255;
Setpoint = Setpoint * 2000 / 255;
if (Setpoint == 0 && Input == 0) Output = 0;

if (Set < In) //SALVAGUARDAR O FACTO DO PID NAO FUNCIONAR
COM VALORES NEGATIVOS
{

    Pfinal = Pteorica - Output;

}
else
{
    Pfinal = Pteorica + Output;

}
if (Pfinal <= 0)
{
    Pfinal = 0;
}
if (Pfinal >= PotMax)
{
    Pfinal = PotMax;
}
duty = Pfinal / PotMax;
if (now - lastMessage > serialPing) { //If it has been long
enough give us

    Serial.print(" || Tsaida:");
    Serial.print(Tres);
    Serial.print(" | Estado SNivel: ");
    Serial.print(digitalRead(snivel));
    Serial.print(" | Estado BombaIN: ");
    Serial.print(digitalRead(bomba));

    Serial.print(" || Duty:");
    Serial.print(duty);

```



```

    Serial.print("\n");
    if (Serial.available() > 0) { //If we sent the program a
command deal
        for (int x = 0; x < 4; x++) {
            switch (x) {
                case 0:
                    kp = Serial.parseFloat();
                    break;
                case 1:
                    ki = Serial.parseFloat();
                    break;
                case 2:
                    kd = Serial.parseFloat();
                    break;
                case 3:
                    for (int y = Serial.available(); y == 0; y--) {
                        Serial.read(); //Clear out any residual junk
                    }
                    break;
            }
        }
        Serial.print(" Kp,Ki,Kd = ");
        Serial.print(kp);
        Serial.print(",");
        Serial.print(ki);
        Serial.print(",");
        Serial.println(kd); //Let us know what we just received
        myPID.SetTunings(kp, ki, kd); //Set the PID gain
constants and start
    }

    lastMessage = now;
}

}

void callback()
{
    if (duty >= 1) {
        digitalWrite(resist, HIGH);
    }
}

```

```

    tempo2 = Ttotal;
}
else if (duty <= 0) {
    digitalWrite(resist, LOW);
    tempo2 = Ttotal;
}
else
{
    if (digitalRead(resist) == 1)
    {
        digitalWrite(resist, LOW);

        tempo2 = Ttotal - (Ttotal * duty);
    }
    else
    {
        digitalWrite(resist, HIGH);
        tempo2 = (Ttotal * duty);

    }

}

}
Timer1.initialize(tempo2);
}

```

Apêndice C

Interface gráfica do ControlDesk

Caso de Estudo 1 - Controlo e Monitorização

Tipo de simulação

1-Virtual

Temperatura desejada

37.000

Temp III

Godot3

+22.300

Temp OUT

Toutfinal

+22.336

Estado da Bomba



Sensor de Nível Max



Caudal III

-0.0008

Caudal OUT

+0.0008

Nível da água no reservatório

Read_P1100rt



Tensões nas entradas e saídas da DS1104

ADC5

-0.0004

ADC6

-0.0004

DAC1

+0.0000

DAC2

+0.0000

I/O 5

+1.88

I/O 8

+0.00

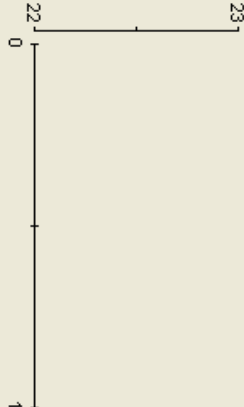
I/O 9

+0.00

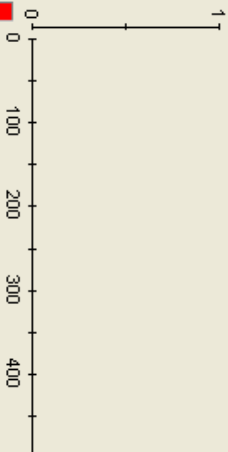
Temp OUT

Plotter

Toutfinal In1

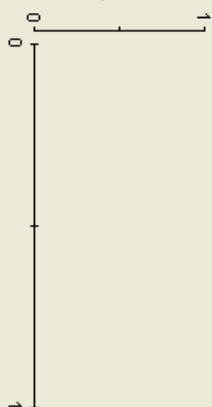


Plotter



Plotter

Scope23 In1



Tool Window

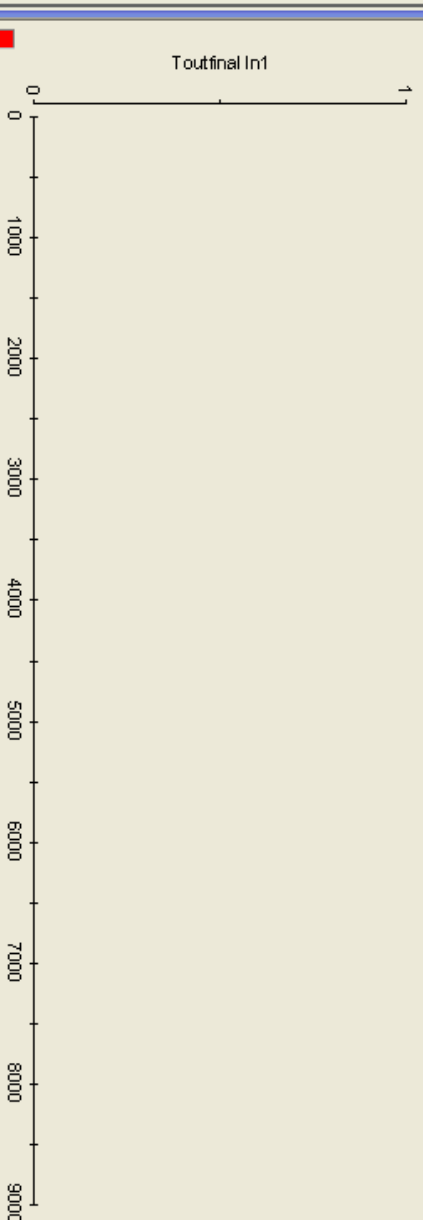
layout12

resetVstet1

Temperatura Saída

39.963

Plotter



Plotter

PPC - teste2 - HostService

Start

Settings...

0 %

Length 9000

Auto Repeat

Downsampling 1000

Trigger Signal

On/Off

Level 0.5

Delay 0

Model Root/Constant10/Value

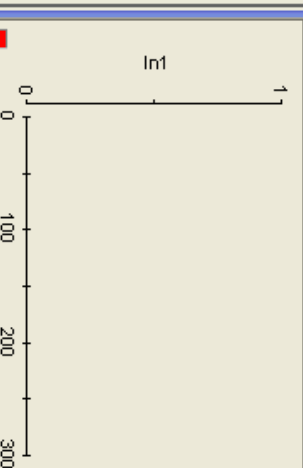
Reference Capture Capture Variables

Take Save... 007 of 007

Caudal de saída
(L/min)

150

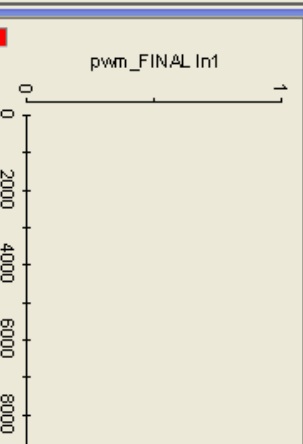
Plotter



PWM entregue ao
sistema

1000

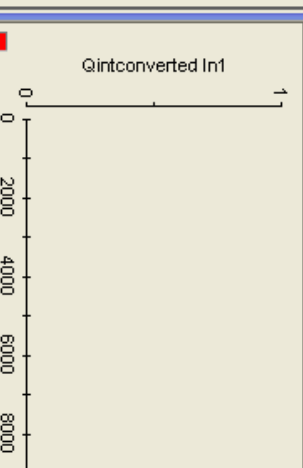
Plotter



Caudal de entrada
(L/min)

1000

Plotter



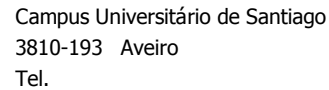
Tool Window

layout12

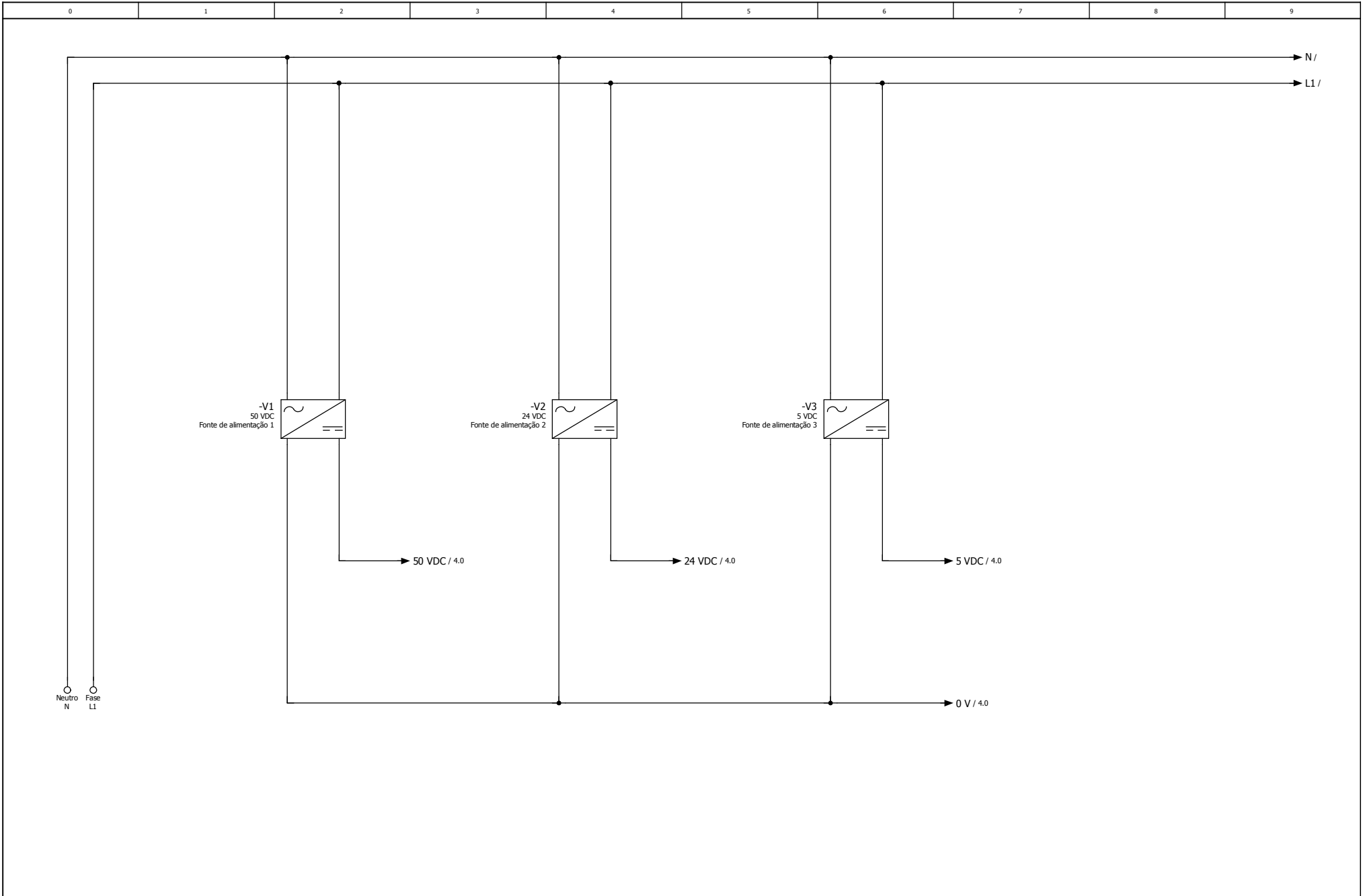
reservtest1

Apêndice D

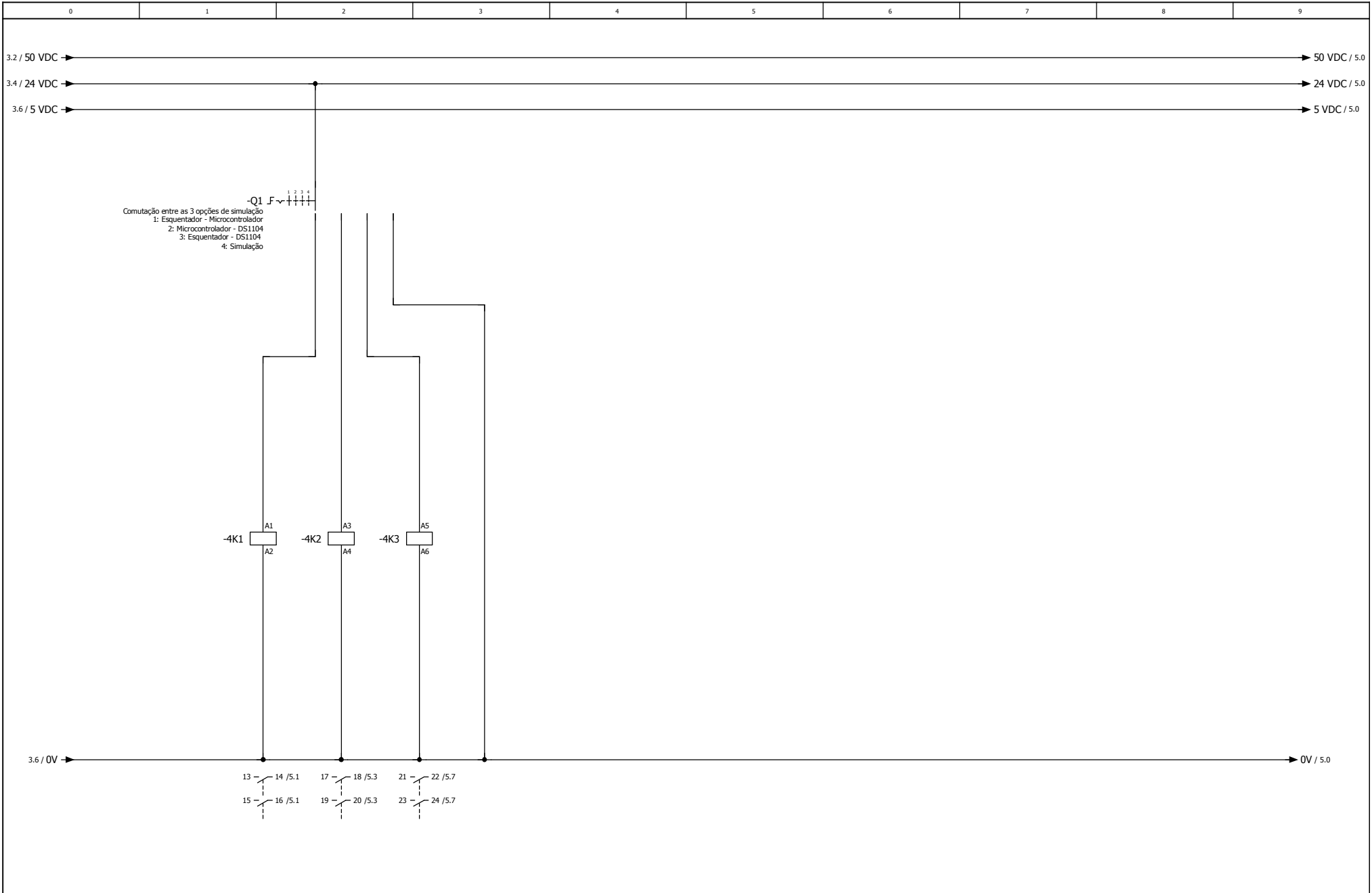
Projeto elétrico do caso de estudo 2

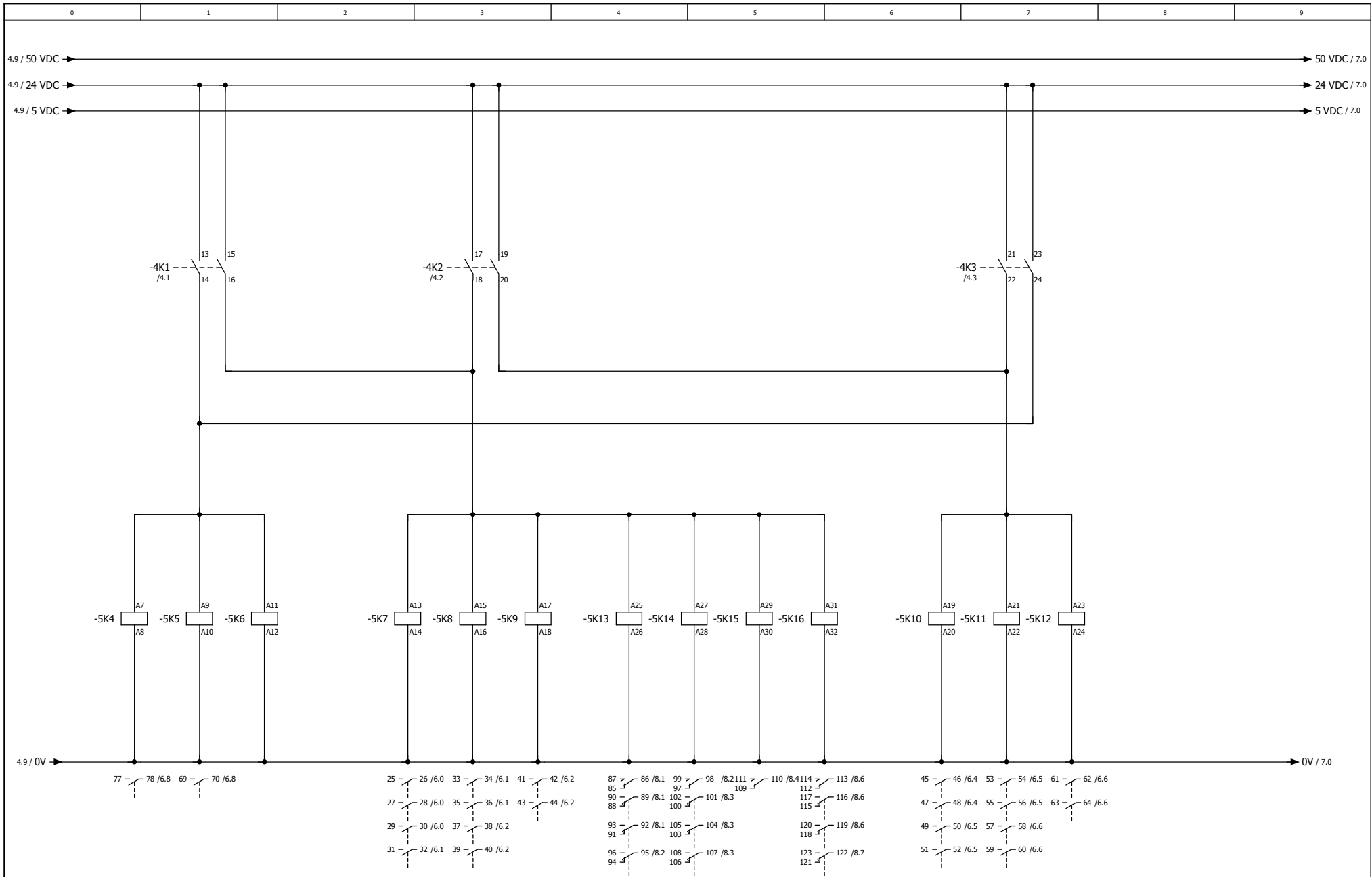


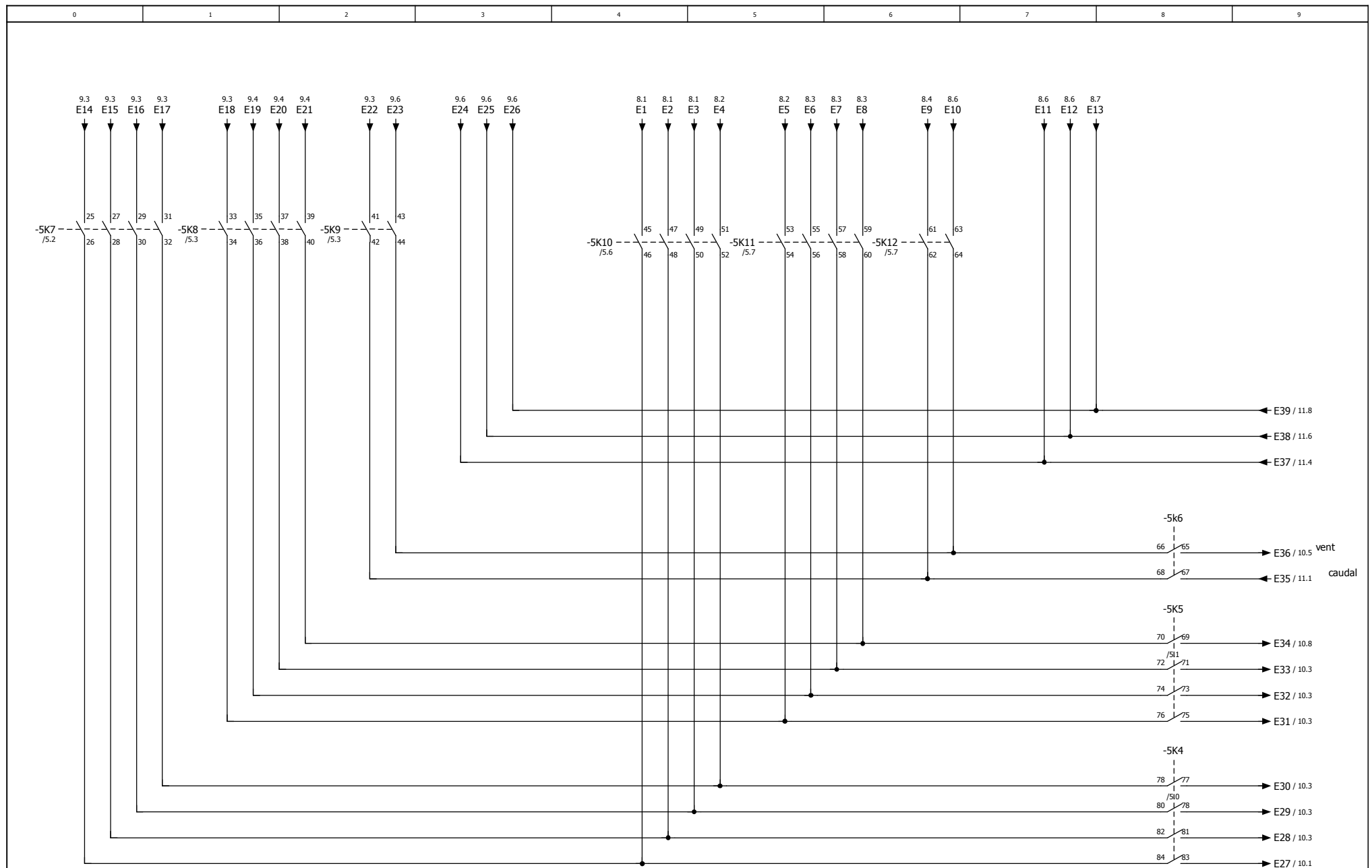
Quantidade de páginas 10

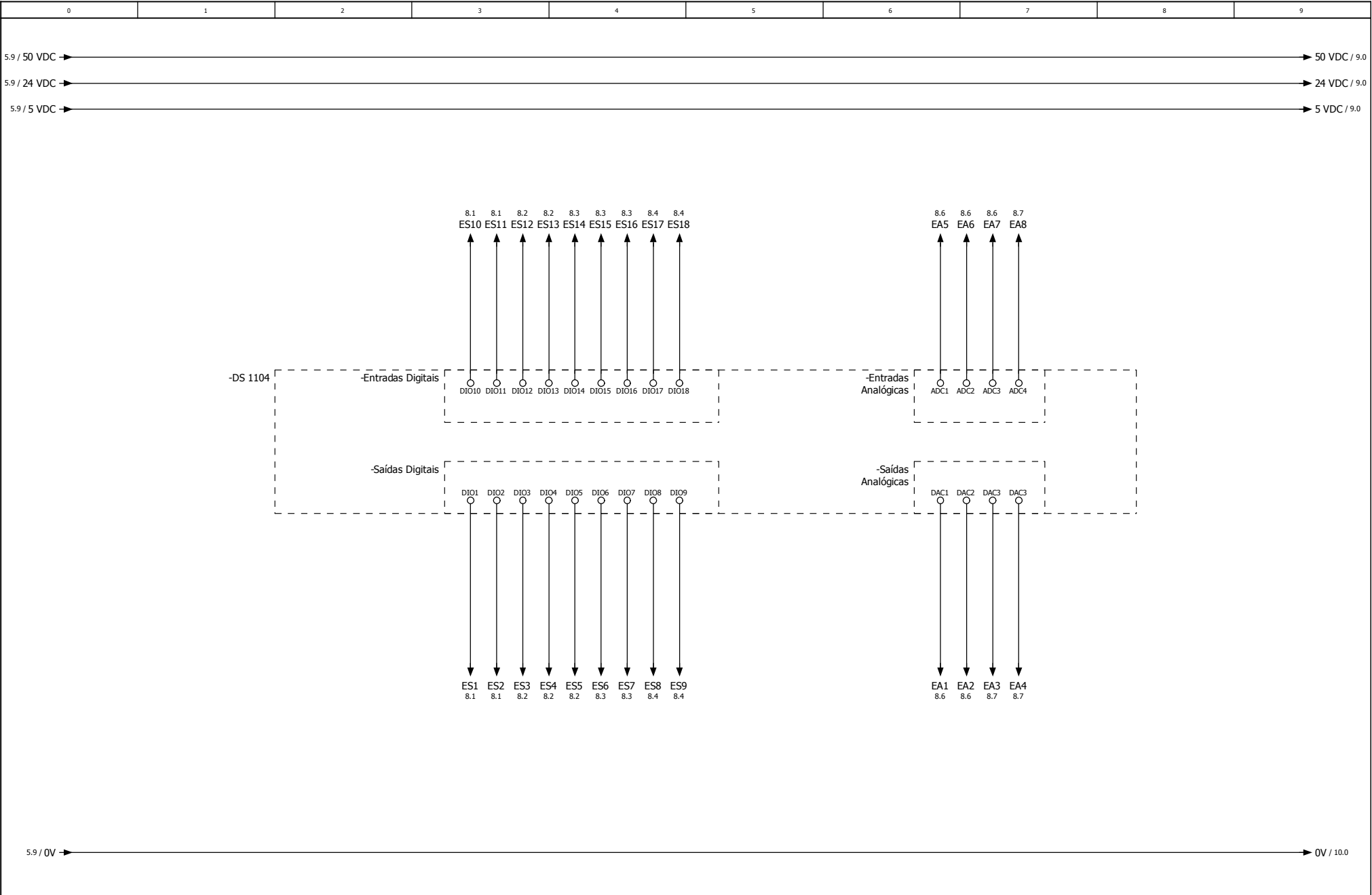


1										4									
			Data	13-07-2016	EPLAN			Universidade de Aveiro	Alimentação				=						
			Editor.	MICHAELPEGO									+						
			Verif		Projeto Elétrico da Bancada de Ensaio														
Alteração	Data	Nome	Orig		Em substituição de		Substituído por			IEC_tpl002		Folha	3						
												Folha	10						





[illegible]





			Data	25-07-2016	EPLAN		Universidade de Aveiro	Comutação de sinais DS1104			=				
			Editor.	MICAELPEGO									+		
			Verif		Projeto Elétrico da Bancada de Ensaios										
Alteração	Data	Nome	Orig		Em substituição de	Substituído por			IEC_tpl002		Folha 8				
											Folha 10				

